Ph.D. version finale

écoulement viscoélastique 2D, modèle rhéologique de Grmela généralisé (tenseur de conformation)

```
      PROGRAM NN2DGR1B
*
*     DEFINITION DES VARIABLES GLOBALES
*     ---------------------------------
*
*     MBLANK ..... ESPACE MEMOIRE DE GESTION DYNAMIQUE
*     ITWO ....... NOMBRE DE MOTS MEMOIRE POUR UN REEL
*     IPRINT ..... NUMERO DE L'UNITE DE SORTIE
*     IA ......... SUPER-TABLEAU DE GESTION DYNAMIQUE
*
*
*     CONSTANTES GLOBALES
*     -------------------
      INTEGER MBLANK
      PARAMETER ( MBLANK = 2500000 )
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER   ITWO
      COMMON  / DPR   / ITWO
      INTEGER IPRINT
      COMMON / DISKS  / IPRINT
      INTEGER   MTOT
      COMMON  / MEMORY / MTOT
      INTEGER   IA
      COMMON  / BLANK / IA(MBLANK)
*
      ITWO = 2
      IPRINT = 2
      MTOT = MBLANK
*
      OPEN(UNIT=1,FILE='c41gr.dat')
      OPEN(UNIT=2,FILE='c41gr.res')
      OPEN(UNIT=3,FILE='estime')
      OPEN(UNIT=4,FILE='estimesu')
      OPEN(UNIT=5,FILE='spyglass.u')
      OPEN(UNIT=6,FILE='spyglass.c')
      OPEN(UNIT=7,FILE='spyglass.p')
      OPEN(UNIT=8,FILE='spyglass.n1')
*
      CALL INDATA
      CALL MODSOL
      CALL OUTPUT
      STOP
      END
*
*     ================================================================
      SUBROUTINE INITBL ()
*     ================================================================
*
*     FONCTION : INITIALISE LE SUPER-TABLEAU IA POUR LA GESTION
*     ---------    DYNAMIQUE DE MEMOIRE
*
*     APPEL    : CALL INITBL ()
*     -----
*
*     VARIABLES GLOBALES
*     ------------------
*
*     TABLE(I)..... LISTE DES TABLEAUX ALLOUES
*     TABTYP(I) ..... TYPE DU TABLEAU TABLE(I)
*                     'INTEGER' OU 'REAL'
*     TBLSIZ(I) ..... TAILLE DU TABLEAU TABLE(I)
*     IBEG(I) ..... ADDRESSE DU PREMIER MOT ENTIER
*                   DU TABLEAU TABLE(I)
*     IEND(I) ..... ADDRESSE DU DERNIER MOT ENTIER
```

```
*                              DU TABLEAU TABLE(I)
*      NTABLE  ..............  NOMBRE DE TABLEAUX
*      NXTTBL  ..............  ADDRESSE DU PREMIER MOT ENTIER
*                              SUIVANT LE DERNIER TABLEAU
*      MAXTBL ...............  NOMBRE MAXIMUM DE TABLEAUX POUVANT
*                              ETRE ALLOUES
*
*      CONSTANTES GLOBALES
*      -------------------
       INTEGER            MAXTBL
       PARAMETER        ( MAXTBL=500 )
*
*      VARIABLES GLOBALES
*      ------------------
       INTEGER           ITWO
       COMMON / DPR    / ITWO
       INTEGER IBEG  , IEND  , TBLSIZ, NTABLE, NXTTBL
       CHARACTER*7 TABLE , ORIGIN
       CHARACTER*7 TABTYP
       COMMON / TABLE1 / TABLE (MAXTBL), TABTYP(MAXTBL), ORIGIN(MAXTBL)
       COMMON / TABLE2 / IBEG  (MAXTBL), IEND  (MAXTBL), TBLSIZ(MAXTBL),
      &                  NTABLE        , NXTTBL
*
*      ================================================================
       NXTTBL = 1
       NTABLE = 0
       RETURN
       END
*
*      ================================================================
       INTEGER FUNCTION ADDTBL (NAME  , LENGTH, TYPE , FROM  )
*      ================================================================
*
*      FUNCTION : ALLOCATION DYNAMIQUE D'UN TABLEAU
*      ---------  RETOURNE L'ADRESSE DU PREMIER ELEMENT DU TABLEAU
*                 NAME
*
*      APPEL :    NX = ADDTBL (NAME  , LENGTH, TYPE , FROM  )
*      -----
*
*      ARGUMENTS
*      ---------
*
*      LENGTH ..............  DIMENSION DU TABLEAU
*      TYPE ................  TYPE DU TABLEAU
*                               = 'INTEGER'
*                               = 'REAL'
*      NAME ................  NOM DU TABLEAU A CREER
*      FROM ................  NOM DE LA SOUS-ROUTINE QUI DEMANDE LA
*                             CREATION DU TABLEAU
*
*      VARIABLES GLOBALES
*      ------------------
       INTEGER           ITWO
       COMMON / DPR    / ITWO
       INTEGER IPRINT
       COMMON / DISKS  / IPRINT
       INTEGER   MTOT
       COMMON  / MEMORY / MTOT
       INTEGER IBEG  , IEND  , TBLSIZ, NTABLE, NXTTBL
       INTEGER            MAXTBL
       PARAMETER        ( MAXTBL=500 )
       CHARACTER*7 TABLE , ORIGIN
       CHARACTER*7 TABTYP
       COMMON / TABLE1 / TABLE (MAXTBL), TABTYP(MAXTBL), ORIGIN(MAXTBL)
       COMMON / TABLE2 / IBEG  (MAXTBL), IEND  (MAXTBL), TBLSIZ(MAXTBL),
```

```
     &                    NTABLE        , NXTTBL
*
*      ARGUMENTS
*      ---------
       INTEGER        LENGTH
       CHARACTER*(*) NAME
       CHARACTER*(*) TYPE
       CHARACTER*(*) FROM
*
*      VARIABLES LOCALES
*      -----------------
       INTEGER LEVEL
       PARAMETER ( LEVEL=1 )
       LOGICAL VF3090
       PARAMETER ( VF3090 = .TRUE. )
       INTEGER I
       INTEGER NWORDS
       INTEGER NSIZE
       CHARACTER*7 TABNAM
       INTEGER     NAMLEN
*
*      ================================================================
       NAMLEN = LEN (NAME)
       TABNAM = '       '
       TABNAM = NAME(1:NAMLEN)
*      ------------------------------------------
*      VERIFIE SI LE TABLEAU NAME A DEJA ETE ALLOUE
*      ------------------------------------------
       DO 100 I = 1,NTABLE
          IF (TABNAM .EQ. TABLE(I)) THEN
             WRITE (IPRINT,1000) TABNAM  , FROM
             CALL LISTBL('ADDTBL')
             STOP
          ENDIF
100    CONTINUE
*      ------------------------------------------------
*      LONGUEUR DU TABLEAU EN MOT ENTIER SUR L'ORDINATEUR
*      ------------------------------------------------
       IF ( TYPE .EQ. 'INTEGER' ) THEN
          NWORDS = LENGTH
       ELSE IF (TYPE .EQ. 'REAL' ) THEN
          NWORDS = LENGTH * ITWO
       ELSE
          WRITE(IPRINT,1050) TABNAM, TYPE
          STOP
       ENDIF
*
       NSIZE = NXTTBL + NWORDS
       IF ( NSIZE .GT. MTOT ) THEN
          WRITE ( IPRINT,1100) TABNAM , NSIZE, MTOT
          STOP
       ELSE
          NTABLE = NTABLE + 1
          IF ( NTABLE .GT. MAXTBL ) THEN
             WRITE(IPRINT,1150) TABNAM, MAXTBL
             STOP
          ENDIF
          TABLE  (NTABLE) = TABNAM
          TBLSIZ (NTABLE) = LENGTH
          TABTYP (NTABLE) = TYPE
          IBEG   (NTABLE) = NXTTBL
          IEND   (NTABLE) = NXTTBL + NWORDS - 1
          ORIGIN (NTABLE) = FROM
          ADDTBL          = NXTTBL
          NXTTBL          = IEND(NTABLE)  + 1
****      ------------------------------------------------------------
```

```
****     TO FORCE DOUBLE WORD ALIGNMENT FOR PROPER VECTORIZATION
****     ON THE IBM-3090 ACTIVATE THE FOLLOWING IF CLAUSE
****     ------------------------------------------------------------
         IF (VF3090) THEN
            IF(TABTYP(NTABLE).EQ.'INTEGER' .AND.MOD(NXTTBL,2).EQ.0) THEN
               NXTTBL = NXTTBL + 1
               IEND(NTABLE) = IEND(NTABLE) + 1
            END IF
         ENDIF
****     ------------------------------------------------------------
****     END OF IF CLAUSE FOR DOUBLE WORD ALIGNMENT ON 3090 VECTOR-F.
****     ------------------------------------------------------------
      END IF
      IF ( DBGLVL .GE. LEVEL ) THEN
         WRITE(IPRINT,1200) NTABLE, TABLE(NTABLE), TBLSIZ(NTABLE),
     &                      TABTYP(NTABLE),NWORDS, IBEG (NTABLE),
     &                      IEND  (NTABLE),ORIGIN(NTABLE)
      ENDIF
      RETURN
*     ------------------------------------------------------------
*     FORMATS
*     ------------------------------------------------------------
1000  FORMAT('*** ADDTBL *** EXECUTION STOPS - TABLE "',
     &           A7,'" HAS ALREADY BEEN ALLOCATED.',/
     &             ,' "ADDTBL" CALLED FROM ROUTINE : ',A7,/
     &          ,'    DUMP OF TABLES ALLOCATED FOLLOWS')
1050  FORMAT('*** ADDTBL *** EXECUTION STOPS - TABLE "',
     &           A7,'" HAS UNKNOWN TYPE =',A10/)
1100  FORMAT ('*** ADDTBL *** ',
     &            'ERROR IN MEMORY ALLOCATION FOR TABLE "',A7,'"',/
     &           ,'    TOTAL SPACE REQUESTED = ',I10,' WORDS,',/
     &           ,'    TOTAL SPACE AVAILABLE = ',I10,' WORDS.',/
     &           ,'    INCREASE STORAGE IN BLANK COMMON'/)
1150  FORMAT('*** ADDTBL *** ',
     &           'EXECUTION STOPS - NOT ENOUGH SPACE IN BOOKEEPING ',
     &           ' ARRAYS TO ALLOCATE TABLE "',A7,'".',/
     &          ,'    INCREASE SIXE OF ARRAY TABLE(I), IBEG(I), ',
     &           'IEND(I), TYPE(I).',/
     &          ,'    CURRENT SIZE IS = MAXTBL =',I5 )
1200  FORMAT('*** ADDTBL *** TABLE',I3,' - "',A7,
     &           '" LENGTH = ',I5,' ',A7,', SIZE = ',I6,' WORDS',
     &           ' FROM IA (',I6,' )',' TO IA (',I6,' ) ORIGIN = ',A7)
*     ------------------------------------------------------------
      END
*
*     ============================================================
      SUBROUTINE RMVTBL (NAME )
*     ============================================================
*
*     FONCTION : ENLEVE LE TABLEAU NAME DU SUPER-TABLEAU IA
*     ----------  PEUT EGALEMENT EFFECTUER LA COMPRESSION DES TABLEAUX
*
*     APPEL :    CALL RMVTBL(NAME)
*     -----
*
*     ARGUMENTS
*     ---------
*
*     NAME ................ NOM DU TABLEAU A ENLEVER
*
*     VARIABLE LOCALE
*     ---------------
*
*     FORCED  ............. OPTION FLAG : (LOGICAL)
*                           = .TRUE. , LE TABLEAU DOIT ETRE
*                                      LE DERNIER DE LA LISTE
```

```
*                          = .FALSE., LE TABLEAU PEUT ETRE
*                                     N'IMPORTE OU
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER     MBLANK
      PARAMETER ( MBLANK =2500000)
      INTEGER     IA
      COMMON      IA ( MBLANK )
      LOGICAL  DEBUG
      INTEGER  DBGLVL
      COMMON  / BUGS   /   DBGLVL, DEBUG
      INTEGER IPRINT
      COMMON / DISKS  / IPRINT
      INTEGER           MAXTBL
      PARAMETER        ( MAXTBL=500 )
      INTEGER IBEG  , IEND  , TBLSIZ, NTABLE, NXTTBL
      CHARACTER*7 TABLE , ORIGIN
      CHARACTER*7 TABTYP
      COMMON / TABLE1 / TABLE (MAXTBL), TABTYP(MAXTBL), ORIGIN(MAXTBL)
      COMMON / TABLE2 / IBEG  (MAXTBL), IEND  (MAXTBL), TBLSIZ(MAXTBL),
     &                  NTABLE        , NXTTBL
*
*     ARGUMENTS
*     ---------
      CHARACTER*(*) NAME
*
*     VARIABLES LOCALES
*     -----------------
      CHARACTER*7 TABNAM
      LOGICAL FORCED
      INTEGER  END   , START, TABSIZ
      INTEGER  I     , J    , NAMLEN
*
      INTEGER LEVEL
      PARAMETER ( LEVEL=1 )
      DATA FORCED / .TRUE. /
*
*     ================================================================
*     RECHERCHE LE TABLEAU A ENLEVER DANS LA LISTE
*     --------------------------------------------
      NAMLEN = LEN(NAME)
      TABNAM = '       '
      TABNAM = NAME(1:NAMLEN)
      DO 300 I = 1,NTABLE
         IF ( TABNAM .EQ. TABLE(I)) THEN
            IF ( FORCED .AND . I .NE. NTABLE ) THEN
               WRITE(IPRINT,1000) TABNAM , I , NTABLE
               CALL LISTBL('RMVTBL')
               STOP
            END IF
            TABSIZ = IEND(I) - IBEG(I) + 1
            START  = IBEG(I)
            END    = IEND(NTABLE)
            NXTTBL = NXTTBL - TABSIZ
            IF ( DBGLVL .GE. LEVEL ) THEN
               WRITE(IPRINT,1100) TABNAM, TBLSIZ(I),TABTYP(I),IBEG(I),
     &                            IEND(I),TABSIZ
               WRITE(IPRINT,1200) NXTTBL
            END IF
*           -------------------------
*           COMPRESSION DES TABLEAUX
*           -------------------------
            IF ( I .LT. NTABLE ) THEN
               DO 100 J = START,END
                  IA(J) = IA(J + TABSIZ)
```

```
100            CONTINUE
*              ------------------------------------
*              AJUSTEMENT DES POINTEURS DES TABLEAUX
*              ------------------------------------
               DO 200 J = I,NTABLE
                  TABLE(J) = TABLE(J+1)
                  IBEG(J)  = IBEG(J+1) - TABSIZ
                  IEND(J)  = IEND(J+1) - TABSIZ
200            CONTINUE
            END IF
            NTABLE = NTABLE - 1
            RETURN
         ENDIF
300   CONTINUE
      WRITE(IPRINT,1300) TABNAM
      CALL LISTBL('RMVTBL')
      STOP
*     ------------------------------------------------------------------
*     FORMATS DES MESSAGES
*     ------------------------------------------------------------------
1000  FORMAT('*** RMVTBL *** ',
     &           'EXECUTION STOPS - TABLE "',A7,'" IS NO. ',I3,
     &           ' IN THE LIST OF ',I3,' TABLES; IS NOT LAST TABLE',/
     &          ,'     DUMP OF TABLES ALLOCATED FOLLOWS')
1100  FORMAT ('*** RMVTBL *** ',
     &           'REMOVAL OF TABLE "',A7,' ',I6,' ',A7,
     &           ' FROM IA(',I6,') , TO IA(',I6,'),',
     &           ' GARBAGE COLLECTION OF ',I6,' WORDS.')
1200  FORMAT ('                ',
     &           'NEXT TABLE TO BE ALLOCATED WILL START AT',
     &           ' NXTTBL = ',I6)
1300  FORMAT('*** RMVTBL *** ',
     &           'EXECUTION STOPS - TABLE "',A7,'" NOT FOUND ',
     &           'DURING TABLE REMOVAL ATTEMPT,'/
     &          ,'     DUMP OF TABLES ALLOCATED FOLLOWS')
*     ------------------------------------------------------------------
      END
*
*     ==================================================================
      INTEGER FUNCTION STOROF (NAME  , FROM )
*     ==================================================================
*
*     FONCTION : RETOURNE LE NOMBRE DE MOTS MEMOIRE OCCUPE
*     ---------  PAR LE TABLEAU NAME : INCLU POSSIBLEMENT
*                LES VIDES NECESSAIRES A L'ALIGNEMENT POUR
*                LE IBM-3090-VF
*
*     APPEL :    NX = STOROF (NAME  , FROM )
*     -----
*
*     ARGUMENTS :
*     ---------
*
*     NAME ................ NOM DU TABLEAU
*     FROM   .............. NOM DE LA SOUS-ROUTINE QUI FAIT L'APPEL
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER          ITWO
      COMMON / DPR   / ITWO
      INTEGER IPRINT
      COMMON / DISKS  / IPRINT
      INTEGER           MAXTBL
      PARAMETER       ( MAXTBL=500 )
      INTEGER IBEG  , IEND  , TBLSIZ, NTABLE, NXTTBL
      CHARACTER*7 TABLE , ORIGIN
```

```
      CHARACTER*7 TABTYP
      COMMON / TABLE1 / TABLE (MAXTBL), TABTYP(MAXTBL), ORIGIN(MAXTBL)
      COMMON / TABLE2 / IBEG  (MAXTBL), IEND  (MAXTBL), TBLSIZ(MAXTBL),
     &                  NTABLE        , NXTTBL
*
*     ARGUMENTS
*     ---------
      CHARACTER*(*) NAME
      CHARACTER*(*) FROM
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER LEVEL
      PARAMETER ( LEVEL=1 )
      INTEGER I
      CHARACTER*7 TABNAM
      INTEGER     NAMLEN , NWORDS
      LOGICAL     FOUND
*
*     ================================================================
      NAMLEN = LEN (NAME)
      TABNAM = '       '
      TABNAM = NAME(1:NAMLEN)
*     ----------------------------
*     VERIFIE SI LE TABLEAU EXISTE
*     ----------------------------
      FOUND = .FALSE.
      DO 100 I = 1,NTABLE
         IF (TABNAM .EQ. TABLE(I)) THEN
            FOUND = .TRUE.
            STOROF = IEND(I) - IBEG(I) + 1
         END IF
100   CONTINUE
      IF ( .NOT. FOUND ) THEN
         WRITE ( IPRINT,1000) NAME, FROM
         STOROF = 0
      END IF
*     ----------------------------------------------------------------
*     FORMATS POUR LES MESSAGES
*     ----------------------------------------------------------------
1000  FORMAT('*** STOROF *** TABLE "',
     &           A7,'" DOES NOT EXIST.',
     &           ' "STOROF" CALLED FROM ROUTINE : ',A7,/ )
*     ----------------------------------------------------------------
      END
*
*     ================================================================
      SUBROUTINE LISTBL (FROM)
*     ================================================================
*
*     FONCTION : FAIT LA LISTE DE TOUS LES TABLEAUX ALLOUES
*     --------
*
*     ARGUMENTS :
*     ---------
*     FROM ............. NOM DE LA SOUS-ROUTINE QUI FAIT L'APPEL
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER IPRINT
      COMMON / DISKS  / IPRINT
      INTEGER           MAXTBL
      PARAMETER       ( MAXTBL=500 )
      INTEGER IBEG  , IEND  , TBLSIZ, NTABLE, NXTTBL
      CHARACTER*7 TABLE , ORIGIN
      CHARACTER*7 TABTYP
```

```
      COMMON / TABLE1 / TABLE (MAXTBL), TABTYP(MAXTBL), ORIGIN(MAXTBL)
      COMMON / TABLE2 / IBEG  (MAXTBL), IEND  (MAXTBL), TBLSIZ(MAXTBL),
     &                  NTABLE        , NXTTBL
*
*     ARGUMENTS
*     ---------
      CHARACTER*(*) FROM
*
*     ================================================================
*      FROM = FROM(1:LEN(FROM))
      WRITE(IPRINT,1000) FROM
      DO 100 I = 1,NTABLE
         WRITE(IPRINT,1100) I,TABLE(I),TBLSIZ(I),TABTYP(I),
     &                      IBEG(I),IEND(I),ORIGIN(I)
100   CONTINUE
      WRITE(IPRINT,1200)
      RETURN
*     ----------------------------------------------------------------
*     FORMATS D'IMPRESSION
*     ----------------------------------------------------------------
1000  FORMAT(//,70('=')/
     &          ,'*** LISTBL *** ',' APPELE DE LA ROUTINE : "',A7,'"',//
     &           ' LISTE DE TOUS LES TABLEAUX ALLOUES A CET INSTANT'/
     &  70('=')/,T5,' NO.',T15,'TABLE',T24,'LONGUEUR',T35,'TYPE',
     &           T47,'DEBUT A',T57,'FIN A',T64,'ORIGINE',/70('='))
1100  FORMAT(T5,I3,' - ',T15,A7,T22,I7,T35,A7,T45,I7,T55,I7,T65,A7)
1200  FORMAT(70('=')/)
*     ----------------------------------------------------------------
      END
*
*     ================================================================
      INTEGER FUNCTION SIZEOF (NAME  , FROM )
*     ================================================================
*
*     FONCTION : CALCULE LA TAILLE DU TABLEAU NAME EN MOTS MEMOIRE
*     ---------
*
*     APPEL :    NX = SIZEOF (NAME  , FROM )
*     -----
*
*     ARGUMENTS
*     ---------
*
*     NAME ................ NOM DU TABLEAU
*     FROM ................ NOM DE LA ROUTINE QUI FAIT L'APPEL
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER          ITWO
      COMMON / DPR   / ITWO
      INTEGER IPRINT
      COMMON / DISKS  / IPRINT
      INTEGER           MAXTBL
      PARAMETER       ( MAXTBL=500 )
      INTEGER IBEG  , IEND  , TBLSIZ, NTABLE, NXTTBL
      CHARACTER*7 TABLE , ORIGIN
      CHARACTER*7 TABTYP
      COMMON / TABLE1 / TABLE (MAXTBL), TABTYP(MAXTBL), ORIGIN(MAXTBL)
      COMMON / TABLE2 / IBEG  (MAXTBL), IEND  (MAXTBL), TBLSIZ(MAXTBL),
     &                  NTABLE        , NXTTBL
*
*     ARGUMENTS
*     ---------
      CHARACTER*(*) NAME
      CHARACTER*(*) FROM
*
```

```
*     VARIABLES LOCALES
*     -----------------
      INTEGER LEVEL
      PARAMETER ( LEVEL=1 )
      INTEGER I
      CHARACTER*7 TABNAM
      INTEGER     NAMLEN , NWORDS
      LOGICAL     FOUND
*
*     ================================================================
      NAMLEN = LEN (NAME)
      TABNAM = '       '
      TABNAM = NAME(1:NAMLEN)
*     --------------------------
*     VERIFIE SI LE TABLEAU EXISTE
*     --------------------------
      FOUND = .FALSE.
      DO 100 I = 1,NTABLE
         IF (TABNAM .EQ. TABLE(I)) THEN
            FOUND = .TRUE.
            IF ( TABTYP(I) .EQ. 'INTEGER') THEN
               SIZEOF = TBLSIZ(I)
            ELSE IF ( TABTYP(I) .EQ. 'REAL') THEN
               SIZEOF = TBLSIZ(I)*ITWO
            END IF
         END IF
100   CONTINUE
      IF ( .NOT. FOUND ) THEN
         WRITE ( IPRINT,1000) NAME, FROM
         SIZEOF = 0
      END IF
*     ----------------------------------------------------------------
*     FORMATS D'IMPRESSION
*     ----------------------------------------------------------------
1000  FORMAT('*** SIZEOF *** TABLE "',
     &           A7,'" DOES NOT EXIST.',
     &           ' "SIZEOF" CALLED FROM ROUTINE : ',A7,/ )
*     ----------------------------------------------------------------
      END
*
*     ================================================================
      INTEGER FUNCTION LOCTBL (NAME  , FROM )
*     ================================================================
*
*     FONCTION : RETOURNE L'ADRESSE DU PREMIER ELEMENT
*     --------   DU TABLEAU 'NAME'
*
*     APPEL :    NX = LOCTBL (NAME  , FROM )
*     -----
*
*     ARGUMENTS :
*     ---------
*     NAME ................ NOM DU TABLEAU
*     FROM ................ NOM DE LA ROUTINE QUI FAIT L'APPEL
*
      IMPLICIT CHARACTER ( A-Z )
*
*     CONSTANTES GLOBALES
*     -------------------
      INTEGER           MAXTBL
      PARAMETER       ( MAXTBL=500 )
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER IPRINT
      COMMON / DISKS  / IPRINT
```

```
      INTEGER IBEG  , IEND  , TBLSIZ, NTABLE, NXTTBL
      CHARACTER*7 TABLE , ORIGIN
      CHARACTER*7 TABTYP
      COMMON / TABLE1 / TABLE (MAXTBL), TABTYP(MAXTBL), ORIGIN(MAXTBL)
      COMMON / TABLE2 / IBEG  (MAXTBL), IEND  (MAXTBL), TBLSIZ(MAXTBL),
     &                  NTABLE        , NXTTBL
*
*     ARGUMENTS
*     ---------
      CHARACTER*(*) NAME
      CHARACTER*(*) FROM
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER LEVEL
      PARAMETER ( LEVEL=1 )
      INTEGER I
      CHARACTER*7 TABNAM
      INTEGER     NAMLEN
      LOGICAL     FOUND
*
*     ================================================================
      NAMLEN = LEN (NAME)
      TABNAM = '       '
      TABNAM = NAME(1:NAMLEN)
*     ----------------------------
*     VERIFIE SI LE TABLEAU EXISTE
*     ----------------------------
      FOUND = .FALSE.
      DO 100 I = 1,NTABLE
         IF (TABNAM .EQ. TABLE(I)) THEN
            FOUND = .FALSE.
            LOCTBL = IBEG(I)
            RETURN
         END IF
100   CONTINUE
      IF ( .NOT. FOUND ) THEN
         WRITE ( IPRINT,1000) NAME, FROM
         LOCTBL = 1
      END IF
*     ----------------------------------------------------------------
*     FORMATS D'IMPRESSION
*     ----------------------------------------------------------------
1000  FORMAT('*** LOCTBL *** TABLE "',
     &           A7,'" DOES NOT EXIST.',
     &           ' "LOCTBL" CALLED FROM ROUTINE : ',A7,/ )
*     ----------------------------------------------------------------
      END
*
*     ================================================================
      SUBROUTINE INDATA
*     ================================================================
*
*     FONCTION : DRIVER D'INPUT DES DONNEES
*     --------
*
*     APPEL : CALL INDATA
*     -----
*
*     DEFINITION DES VARIABLES GLOBALES
*     ---------------------------------
*
*     IDIM ......... TYPE DE PROBLEME
*          ......... ----------------
*          ......... = 0 : 2-D
*          ......... = 1 : AXISYMETRIQUE
```

```
*           ......... = 2 : AXISYMETRIQUE + TOURBILLON
*           ......... = 3 : 3-D
*      ITEMP ........ THERMIQUE
*            ........ ---------
*            ........ = 0 SANS THERMIQUE
*            ........ = 1 AVEC THERMIQUE
*      NCOORD ....... NOMBRE DE COORDONNEES
*      NUMNP ........ NOMBRE TOTAL DE NOEUDS DANS LE MAILLAGE
*      NELEM ........ NOMBRE TOTAL D'ELEMENTS
*      NGROUP ....... NOMBRE DE GROUPES D'ELEMENTS
*      NNPE ......... NOMBRE DE NOEUDS PAR ELEMENTS
*      NDEP ......... NOMBRE DE VARIABLES DEPENDANTES
*      EPS .......... INVERSE DU FACTEUR DE PENALISATION
*      IFIX(I) ...... IFIX(1) = NFIXU : NOMBRE DE CONDITIONS LIMITES
*                                       NON NULLES POUR U
*                     IFIX(2) = NFIXV : IDEM POUR V
*                     IFIX(3) = NFIXW : IDEM POUR W
*                     IFIX(4) = NFIXT : IDEM POUR T
*      IFIX1(I) ..... IFIX1(1)= NFIXCXX
*                     IFIX1(2)= NFIXCYY
*                     IFIX1(3)= NFIXCXY
*      NFIXB ........ NOMBRE DE CONDITIONS LIMITES NON NULLES (VITESSES ET T)
*      NFIXB1 ....... NOMBRE DE CONDITIONS LIMITES NON NULLES (CONFORMATION)
*      LDOF(4) ...... NUMEROTATION DES VARIABLES ACTIVES
*      KDOF(4) ...... NUMERO DES VARIABLES PRINCIPALES DANS LE
*                     TABLEAU COMPACT
*      MELDOF ....... DEGRE DE LIBERTE MAXIMAL D'UN ELEMENT (NNPE*NDEP)
*      MELDOF1 ...... DDL MAXIMUM D'UN ELEMENT EN CONFORMATION (NNPE1*NDEP1)
*      NEQT ......... NOMBRE TOTAL D'EQUATIONS (VITESSES ET T)
*      NEQT1 ........ NOMBRE TOTAL D'EQUATIONS (CONFORMATION)
*      ESPMAT ....... ESPACE MATRICIEL DU SYSTEME GLOBAL (VITESSES ET T)
*      ESPMAT1 ...... ESPACE MATRICIEL DU SYSTEME GLOBAL (CONFORMATION)
*
*
*      VARIABLES GLOBALES
*      ------------------
       INTEGER MBLANK, MAXVAR
       PARAMETER ( MBLANK = 2500000 , MAXVAR = 4 )
       INTEGER IA
       COMMON / BLANK / IA(MBLANK)
       INTEGER IPRINT
       COMMON / DISKS  / IPRINT
       INTEGER IDIM, ITEMP, NCOORD
       COMMON / CONTRL1 / IDIM, ITEMP, NCOORD
       INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
       COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
      &                   NUMNP1, NNPE1, NDEP1
       INTEGER NFIXB, IFIX, NFIXB1, IFIX1
       COMMON / FIX / NFIXB, IFIX(MAXVAR), NFIXB1, IFIX1(3)
       INTEGER NEQT, MELDOF, ESPMAT
       COMMON / MATDIM / NEQT, MELDOF, ESPMAT
       INTEGER NEQT1, MELDOF1, ESPMAT1
       COMMON / MATDIM1 / NEQT1, MELDOF1, ESPMAT1
       REAL*8 EPS
       COMMON / EPS / EPS
*
*      VARIABLES LOCALES
*      -----------------
       INTEGER NNCOORD, NIDEQ, NCONSTR, NID, NCONNEC, NDIAG, NLMGLOB
       INTEGER NNCOORD1,NIDEQ1,NCONSTR1,NID1,NCONNEC1,NDIAG1,NLMGLOB1
       INTEGER ADDTBL, LOCTBL , SIZEOF , STOROF
*
*      ================================================================
*      LECTURE DES VARIABLES DE CONTROLE
*      ---------------------------------
       CALL CONTRL
```

```
*     ------------------------------
*     ALLOCATION DES TABLEAUX DANS IA
*     ------------------------------
      CALL INITBL()
      NNCOORD = ADDTBL( 'COORD',  NUMNP*NCOORD,  'REAL',    'INDATA')
      NNCOORD1= ADDTBL( 'COORD1', NUMNP1*NCOORD, 'REAL',    'INDATA')
      NIDEQ   = ADDTBL( 'IDEQ',   NUMNP*NDEP,    'INTEGER', 'INDATA')
      NIDEQ1  = ADDTBL( 'IDEQ1',  NUMNP1*NDEP1,  'INTEGER', 'INDATA')
      NCONSTR = ADDTBL( 'CONSTR', NFIXB,         'REAL',    'INDATA')
      NCONSTR1= ADDTBL( 'CONSTR1',NFIXB1,        'REAL',    'INDATA')
      NID     = ADDTBL( 'ID',     NUMNP*NDEP,    'INTEGER', 'INDATA')
      NID1    = ADDTBL( 'ID1',    NUMNP1*NDEP1,  'INTEGER', 'INDATA')
      CALL LISTBL('INDATA')
      CALL INNODE( IA(NID), IA(NNCOORD), 0, NUMNP, NDEP )
      CALL INNODE( IA(NID1),IA(NNCOORD1),1, NUMNP1,NDEP1)
      CALL EQNUM( IA(NID), IA(NIDEQ), 0, NUMNP, NDEP )
      CALL EQNUM( IA(NID1),IA(NIDEQ1),1, NUMNP1,NDEP1)
      CALL PRTNOD( IA(NID), IA(NIDEQ), IA(NNCOORD), 0, NUMNP, NDEP )
      CALL PRTNOD( IA(NID1),IA(NIDEQ1),IA(NNCOORD1),1, NUMNP1,NDEP1)
      CALL INCONS( IA(NCONSTR), IA(NID), IA(NIDEQ), 0, NUMNP, NDEP,
     &             NFIXB )
      CALL INCONS( IA(NCONSTR1),IA(NID1),IA(NIDEQ1),1, NUMNP1,NDEP1,
     &             NFIXB1)
      CALL PRTCON( IA(NCONSTR), IA(NIDEQ), 0, NUMNP, NDEP, NFIXB )
      CALL PRTCON( IA(NCONSTR1),IA(NIDEQ1),1, NUMNP1,NDEP1,NFIXB1)
      CALL RMVTBL('ID1')
      CALL RMVTBL('ID')
      NCONNEC = ADDTBL('CONNEC', NELEM*NNPE,    'INTEGER', 'INDATA')
      NCONNEC1= ADDTBL('CONNEC1',NELEM*NNPE1,   'INTEGER', 'INDATA')
      NLMGLOB = ADDTBL('LMGLOB', NELEM*MELDOF,  'INTEGER', 'INDATA')
      NLMGLOB1= ADDTBL('LMGLOB1',NELEM*MELDOF1,'INTEGER', 'INDATA')
      NDIAG   = ADDTBL('DIAG',   NEQT,          'INTEGER', 'INDATA')
      NDIAG1  = ADDTBL('DIAG1',  NEQT1,         'INTEGER', 'INDATA')
      NLCIEL  = ADDTBL('LCIEL',  NEQT,          'INTEGER', 'INDATA')
      NLCIEL1 = ADDTBL('LCIEL1', NEQT1,         'INTEGER', 'INDATA')
      CALL LISTBL('INDATA')
      CALL INELEM( IA(NCONNEC), IA(NLCIEL), IA(NDIAG), IA(NLMGLOB),
     &             IA(NIDEQ), 0, NUMNP, NDEP, NEQT, MELDOF, NNPE )
      CALL INELEM( IA(NCONNEC1),IA(NLCIEL1),IA(NDIAG1),IA(NLMGLOB1),
     &             IA(NIDEQ1),1, NUMNP1,NDEP1,NEQT1,MELDOF1,NNPE1)
      CALL PRTTAB( IA(NCONNEC), IA(NDIAG), IA(NLCIEL), 0, NNPE, NEQT )
      CALL PRTTAB( IA(NCONNEC1),IA(NDIAG1),IA(NLCIEL1),1, NNPE1,NEQT1)
*     CALL PRTLMG( IA(NLMGLOB) )
      CALL RMVTBL('LCIEL1')
      CALL RMVTBL('LCIEL')
      CALL LISTBL('INDATA')
      RETURN
      END
*     ================================================================
      SUBROUTINE CONTRL
*     ================================================================
*
*     FONCTION : LECTURE DES VARIABLES DE CONTROLE
*     --------
*
*     APPEL : CALL CONTRL
*     -----
*
*     DEFINITION DES VARIABLES DE CONTROLE
*     ------------------------------------
*
*     MAXITER ..... NOMBRE MAXIMAL D'ITERATIONS (MOUVEMENT)
*     MAXITER1 ..... NOMBRE MAXIMAL D'ITERATIONS (CONFORMATION)
*     MAXITGL ..... NOMBRE MAXIMAL D'ITERATIONS VISCOELASTIQUES
*     UPWIND ..... = 0 PAS D'UPWINDING (GALERKIN)
*            ..... = 1 UPWINDING INCONSISTANT (SU)
```

```
*              ....... = 2 UPWINDING CONSISTANT (SUPG)
*      NEWTON ....... = 0 PREMIERE ITERATION NEWTONNIENNE
*              ....... = 1 PREMIERE ITERATION VISCOELASTIQUE (A DE PLUS FAIBLE)
*
*      VARIABLES GLOBALES
*      ------------------
       INTEGER MAXVAR
       PARAMETER ( MAXVAR = 4 )
       INTEGER IPRINT
       COMMON / DISKS / IPRINT
       INTEGER IDIM, ITEMP, NCOORD
       COMMON / CONTRL1 / IDIM, ITEMP, NCOORD
       INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
       COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
      &                   NUMNP1, NNPE1, NDEP1
       INTEGER NFIXB, IFIX, NFIXB1, IFIX1
       COMMON / FIX / NFIXB, IFIX(MAXVAR), NFIXB1, IFIX1(3)
       INTEGER LDOF, KDOF
       COMMON / DOF / LDOF(MAXVAR), KDOF(MAXVAR)
       INTEGER NEQT, MELDOF, ESPMAT
       COMMON / MATDIM / NEQT, MELDOF, ESPMAT
       INTEGER NEQT1, MELDOF1, ESPMAT1
       COMMON / MATDIM1 / NEQT1, MELDOF1, ESPMAT1
       REAL*8 EPS
       COMMON / EPS / EPS
       INTEGER MAXITER, MAXITER1, MAXITGL
       COMMON / ITER / MAXITER, MAXITER1, MAXITGL
       INTEGER UPWIND
       COMMON / UPWINDING / UPWIND
       INTEGER NEWTON
       COMMON / PREMIERE / NEWTON
*
*      VARIABLES LOCALES
*      -----------------
       CHARACTER*75 TITRE, TITRE1,LIGNE
       INTEGER NFIXU, NFIXV, NFIXW, NFIXT, IDOF(MAXVAR)
       INTEGER NFIXCXX, NFIXCYY, NFIXCXY
       EQUIVALENCE (NFIXU, IFIX(1)), (NFIXV, IFIX(2)),
      &            (NFIXW, IFIX(3)), (NFIXT, IFIX(4))
       EQUIVALENCE (NFIXCXX, IFIX1(1)), (NFIXCYY, IFIX1(2)),
      &            (NFIXCXY, IFIX1(3))
*
*      ================================================================
*      INITIALISATION DE IDOF, LDOF, KDOF
*      ----------------------------------
       DO 5 I = 1, MAXVAR
         IDOF(I) = 0
         LDOF(I) = 0
         KDOF(I) = 0
5      CONTINUE
*      ---------------------------------
*      LECTURE DES VARIABLES DE CONTROLE
*      ---------------------------------
       READ(1,1001) TITRE
       READ(1,1001) TITRE1
       READ(1,1001) LIGNE
       READ(1,*) IDIM, ITEMP
       READ(1,1001) LIGNE
       READ(1,*) NUMNP, NELEM, NGROUP, NNPE,
      &            NFIXU, NFIXV, NFIXW, NFIXT, EPS, MAXITER
       READ(1,1001) LIGNE
       READ(1,*) NUMNP1, NFIXCXX, NFIXCYY, NFIXCXY, MAXITER1, MAXITGL,
      &          UPWIND, NEWTON
       NFIXB = NFIXU + NFIXV + NFIXW + NFIXT
       NFIXB1= NFIXCXX + NFIXCYY + NFIXCXY
*      ------------------------------------
```

```
*     VALIDATION DE IDIM ET CALCUL DE NCOORD
*     --------------------------------------
      IF ( IDIM.EQ.0 .OR. IDIM.EQ.1) THEN
        NCOORD = 2
        IDOF(3) = 1
      ELSEIF ( IDIM.EQ.2 .OR. IDIM.EQ.3) THEN
        NCOORD = 3
      ELSE
        WRITE(IPRINT,1200)
        WRITE(IPRINT,1300) IDIM
        STOP
      ENDIF
*     ------------------
*     VALIDATION DE ITEMP
*     ------------------
      IF ( ITEMP.EQ.0 ) THEN
        IDOF(4) = 1
      ELSEIF ( ITEMP.NE.1 ) THEN
        WRITE(IPRINT,1200)
        WRITE(IPRINT,1400) ITEMP
        STOP
      ENDIF
*     --------------------
*     VALIDATION DE UPWIND
*     --------------------
      IF( UPWIND.NE.0.AND.UPWIND.NE.1.AND.UPWIND.NE.2)  THEN
         WRITE(IPRINT,1200)
         WRITE(IPRINT,1500) UPWIND
         STOP
      ENDIF
*     ----------
*     VALIDATION DE NEWTON
*     --------------------
      IF( NEWTON.NE.0.AND.NEWTON.NE.1 ) THEN
         WRITE(IPRINT,1200)
         WRITE(IPRINT,1501) NEWTON
         STOP
      ENDIF
*     ------------------------------------
*     NUMEROTATION DES VARIABLES DEPENDANTES
*     ------------------------------------
      NDEP = 0
      DO 100 I = 1, MAXVAR
        IF ( IDOF(I).EQ.0 ) THEN
          NDEP = NDEP + 1
          LDOF(I) = NDEP
          KDOF(NDEP) = I
        ENDIF
100   CONTINUE
      MELDOF = NNPE*NDEP
      NNPE1  = 4
      NDEP1  = 3
      MELDOF1= NNPE1*NDEP1
*
*     IMPRESSION DES VARIABLES DE CONTROLE
*     ------------------------------------
      WRITE(IPRINT,1001) TITRE
      WRITE(IPRINT,1002)
      WRITE(IPRINT,1000) IDIM,ITEMP
      WRITE(IPRINT,1100) NUMNP, NELEM, NGROUP, NNPE,
     &               IFIX(1), IFIX(2), IFIX(3), IFIX(4), EPS, MAXITER
      WRITE(IPRINT,1101) NUMNP1, IFIX1(1), IFIX1(2), IFIX1(3), MAXITER1,
     &                   MAXITGL, UPWIND, NEWTON
      WRITE(IPRINT,1002)
*     ------------------------------------------------------------
*     FORMAT DE LECTURE ET MESSAGES D'ERREUR
```

```
*     ------------------------------------------------------------------
1001  FORMAT(A75)
1002  FORMAT(70('='))
1000  FORMAT(7HIDIM  =,I3,5X,7HITEMP =,I3)
1100  FORMAT(7HNUMNP =,I4,4X,7HNELEM =,I3,5X,7HNGROUP=,I3,5X,
     &       7HNNPE  =,I3,/,7HNFIXU =,I3,5X,7HNFIXV =,I3,5X,7HNFIXW =,
     &       I3,5X,7HNFIXT =,I3,/,'EPS   =',E6.1,2X,'MAXITER=',I4 /)
1101  FORMAT('NUMNP1  =',I3,/,'NFIXCXX =',I3,3X,'NFIXCYY =',I3,3X,
     &       'NFIXCXY =',I3,/,'MAXITER1=',I3,3X,'MAXITGL =',I3,3X,
     &       'UPWIND  =',I3,3X,'NEWTON =',I3 /  )
1200  FORMAT(/,'ERREUR DANS "CONTRL"  *** ARRET DU PROGRAMME ***'
     &      ,/60('='))
1300  FORMAT('LA VARIABLE IDIM DOIT ETRE COMPRISE ENTRE 0 ET 3',
     &       /60('=')/,'  VALEUR LUE :  IDIM = ',I2)
1400  FORMAT('LA VARIABLE ITEMP DOIT ETRE EGALE A 1 OU 0',/,
     &       60('=')/,'  VALEUR LUE : ITEMP = ',I2)
1500  FORMAT('LA VARIABLE UPWIND DOIT ETRE EGALE A 0 OU 1 OU 2',/,
     &       60('=')/,'  VALEUR LUE : UPWIND = ',I2)
1501  FORMAT('LA VARIABLE NEWTON DOIT ETRE EGALE A 1 OU 0',/,
     &       60('=')/,'  VALEUR LUE : NEWTON = ',I2)
*
*     ------------------------------------------------------------------
      RETURN
      END
*
*     ==================================================================
      SUBROUTINE INNODE( ID, COORD, FLAG, NUMN, NDE )
*     ==================================================================
*
*     FONCTION : LECTURE DES TABLEAUX ID ( NUMNP, NDEP) ET
*     --------   COORD ( NUMNP, NCOORD )
*
*     APPEL : CALL INNODE ( IA(NID), IA(NNCOORD), 0 OU 1, NUMNP, NDEP)
*     -----
*
*     ARGUMENTS
*     ---------
*
*     ID ................. TABLEAU DES CODES DE CONDITIONS LIMITES
*     COORD .............. TABLEAU DES COORDONNEES DES NOEUDS
*     FLAG ............... 0 => EQUATIONS DU MOUVEMENT
*                          1 => EQUATIONS DE CONFORMATION
*     NUMN ............... NOMBRE DE NOEUDS (MAILLAGE)
*     NDE ................ NOMBRE DE VARIABLES DEPENDANTES
*
*     DEFINITION DES VARIABLES LOCALES
*     --------------------------------
*
*     IDTEMP(MAXVAR) ......... TABLEAU TEMPORAIRE DES CODES DE C.L.
*     CFIX(MAXVAR) ........... NOMBRE DE CONDITIONS LIMITES PAR COLONNE
*     NOEUD .................. NUMERO DU NOEUD
*     INOEUD ................. COMPTEUR (NUMERO DU NOEUD)
*     UCARACT ................ VITESSE CARACTERISTIQUE DE L'ECOULEMENT
*     HCARACT ................ LONGEUR CARACTERISTIQUE DE LA GEOMETRIE
*     LAMBDA ................. TEMPS CARACTERISTIQUE DU FLUIDE
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER MAXVAR
      PARAMETER ( MAXVAR = 4 )
      INTEGER IPRINT
      COMMON / DISKS / IPRINT
      INTEGER IDIM, ITEMP, NCOORD
      COMMON / CONTRL1 / IDIM, ITEMP, NCOORD
      INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
      COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
```

```
     &                    NUMNP1, NNPE1, NDEP1
      INTEGER LDOF, KDOF
      COMMON / DOF / LDOF(MAXVAR), KDOF(MAXVAR)
      INTEGER NFIXB, IFIX, NFIXB1, IFIX1
      COMMON / FIX / NFIXB, IFIX(MAXVAR), NFIXB1, IFIX1(3)
      REAL*8 DE
      COMMON / DEBORAH / DE
      REAL*8 XMAX
      COMMON / SORTIE / XMAX
*
*     ARGUMENTS
*     ---------
      INTEGER FLAG, NUMN, NDE
      INTEGER ID(NUMN,NDE)
      REAL*8 COORD(NUMN,NCOORD)
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER I, IDTEMP(MAXVAR), NOEUD, CFIX(MAXVAR), NBVAR
      REAL*8 UCARACT, HCARACT, LAMBDA
      CHARACTER*75 LIGNE
*
*     INITIALISATION
*
      DO 25 I=1, MAXVAR
        CFIX(I) = 0
25    CONTINUE
*     ================================================================
*     LECTURE DES PARAMETRES CARACTERISTIQUES DE L'ECOULEMENT
*     ET DU FLUIDE
*     ---------------------------------------------------------
      IF(FLAG.EQ.0) THEN
        NBVAR = MAXVAR
        READ(1,1400) LIGNE
        READ(1,*) UCARACT, HCARACT, LAMBDA, XMAX
        DE = LAMBDA*UCARACT/HCARACT
        WRITE(IPRINT,2000) DE
        WRITE(IPRINT,2001) XMAX
        XMAX=XMAX/HCARACT
      ELSEIF(FLAG.EQ.1) THEN
        NBVAR = 3
      ELSE
        WRITE(IPRINT,1800) FLAG
        STOP
      ENDIF
*
*
*     ================================================================
*     LECTURE DES CODES DE CONDITIONS LIMITES ET DES COORDONNEES
*     ----------------------------------------------------------
      READ(1,1400) LIGNE
      DO 100 INOEUD = 1, NUMN
        READ(1,*)  NOEUD, ( IDTEMP(J), J=1,NBVAR ),
     &             ( COORD(INOEUD,J), J=1,NCOORD)
*     ================================================================
*     ADIMENSIONNALISATION DU MAILLAGE
*     --------------------------------
        DO 110 J=1,NCOORD
          COORD(INOEUD,J) = COORD(INOEUD,J)/HCARACT
 110    CONTINUE
*       ----------------------------
*       VALIDATION DU NUMERO DU NOEUD
*       ----------------------------
        IF ( NOEUD.LT.1 .OR. NOEUD.GT.NUMN ) THEN
          WRITE(IPRINT,1050)
          WRITE(IPRINT,1100) NOEUD
```

```
          STOP
        ENDIF
*       ---------------------------------------
*       VALIDATION DU CODE DE CONDITION LIMITE
*       ---------------------------------------
        DO 200 J = 1, NBVAR
          IF ( IDTEMP(J).LT.0 .OR. IDTEMP(J).GT.2 ) THEN
            WRITE(IPRINT,1050)
            WRITE(IPRINT,1200) NOEUD, J, IDTEMP(J)
            STOP
          ELSEIF ( IDTEMP(J).EQ.2 ) THEN
            CFIX(J) = CFIX(J) + 1
          ENDIF
*         ---------------------------------------------
*         TRANSFERT DE IDTEMP VERS ID APRES VERIFICATION
*         ---------------------------------------------
          IF(FLAG.EQ.0) THEN
            IF ( LDOF(J).NE.0 ) ID(NOEUD, LDOF(J)) = IDTEMP(J)
          ELSE
            ID(NOEUD,J) = IDTEMP(J)
          ENDIF
200     CONTINUE
100   CONTINUE
*     ------------------------------------------------------
*     VALIDATION DU NOMBRE DE CONDITION LIMITE PAR COLONNE
*     ------------------------------------------------------
      DO 300, I = 1, NBVAR
        IF(FLAG.EQ.0) THEN
          IF ( IFIX(I).NE.CFIX(I) ) THEN
            WRITE(IPRINT,1050)
            WRITE(IPRINT,1300) I, CFIX(I), IFIX(I)
            STOP
          ENDIF
        ELSE
          IF(IFIX1(I).NE.CFIX(I)) THEN
            WRITE(IPRINT,1050)
            WRITE(IPRINT,1301) I, CFIX(I), IFIX1(I)
            STOP
          ENDIF
        ENDIF
300   CONTINUE
*     ------------------------------------------------------------
*     FORMAT DE LECTURE ET MESSAGE D'ERREUR
*     ------------------------------------------------------------
1000  FORMAT( I5, 4(I3), T31, 3(E10.6) )
1050  FORMAT(/,'ERREUR DANS "INNODE"  *** ARRET DU PROGRAMME ***'/
     &       60('=')/)
1100  FORMAT('LE NUMERO DU NOEUD EST HORS LIMITE',/60('=')/,
     &       '  VALEUR LUE : # NOEUD = ',I5,/,
     &       '  DOMAINE POSSIBLE [ 0 ,',I5,']')
1200  FORMAT('LE CODE DE CONDITION LIMITE DOIT ETRE COMPRIS ENTRE',
     &       ' 0 ET 2',/60('=')/'  VALEUR LUE : # NOEUD = ',I5,
     &       ' # COL = ',I2,' CODE = ',I2)
1300  FORMAT('LE NOMBRE DE CONDITIONS LIMITES DANS LA',
     &       ' COLONNE ',I2,/,'NE CORRESPOND PAS A LA',
     &       ' VALEUR LUE DANS NFIX',/60('=')/,'  DANS "ID" : ',
     &       '    # COND.= ',I3,/,'  DANS "CONTRL" : # COND.= ',I3)
1301  FORMAT('LE NOMBRE DE CONDITIONS LIMITES DANS LA',
     &       ' COLONNE ',I2,/,'NE CORRESPOND PAS A LA',
     &       ' VALEUR LUE DANS NFIX1',/60('=')/,'  DANS "ID1": ',
     &       '    # COND.= ',I3,/,'  DANS "CONTRL" : # COND.= ',I3)

1400  FORMAT(A75)
1800  FORMAT(/,'ERREUR DANS "INNODE" *** ARRET DU PROGRAMME ***',/,
     &         'FLAG =',I3,' DOIT ETRE 0 OU 1')
2000  FORMAT(' NOMBRE DE DEBORAH     = ',F10.4)
```

```
2001  FORMAT(' LONGEUR DE LA FILIERE = ',F10.4, ' CM')
*     ------------------------------------------------------------------
      RETURN
      END
*
*     ==================================================================
      SUBROUTINE EQNUM( ID, IDEQ, FLAG, NUMN, NDE )
*     ==================================================================
*
*     FONCTION : NUMEROTATION DES EQUATIONS ( GENERE IDEQ )
*     ---------
*
*     APPEL : CALL EQNUM ( IA(NID), IA(NIDEQ), 0 OU 1, NUMNP, NDEP )
*     -----
*
*     ARGUMENTS
*     ---------
*
*     ID ................. TABLEAU DES CODES DE CONDITIONS LIMITES
*     IDEQ ................ TABLEAU DE NUMEROTATION DES EQUATIONS
*     FLAG ................ 0 => EQUATIONS DU MOUVEMENT
*          ................ 1 => EQUATIONS DE CONFORMATION
*     NUMN ................ NOMBRE DE NOEUDS
*     NDE ................. NOMBRE DE VARIABLES DEPENDANTES

*     DEFINITION DES VARIABLES LOCALES
*     --------------------------------
*
*     NEQ ................. COMPTEUR (NUMERO DES EQUATIONS : CODE 0)
*     NFIX ................ COMPTEUR (NUMERO DES EQUATIONS : CODE 2)
*     IE .................. VARIABLE INTERMEDIARE = ID(I,J)
*
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER MAXVAR
      PARAMETER ( MAXVAR = 4 )
      INTEGER IPRINT
      COMMON / DISKS / IPRINT
      INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
      COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
     &                   NUMNP1, NNPE1, NDEP1
      INTEGER LDOF, KDOF
      COMMON / DOF / LDOF(MAXVAR), KDOF(MAXVAR)
      INTEGER NFIXB, IFIX, NFIXB1, IFIX1
      COMMON / FIX / NFIXB, IFIX(MAXVAR), NFIXB1, IFIX1(3)
      INTEGER NEQT, MELDOF, ESPMAT
      COMMON / MATDIM / NEQT, MELDOF, ESPMAT
      INTEGER NEQT1, MELDOF1, ESPMAT1
      COMMON / MATDIM1 / NEQT1, MELDOF1, ESPMAT1
*
*     ARGUMENTS
*     ---------
      INTEGER FLAG, NUMN, NDE
      INTEGER IDEQ(NUMN, NDE)
      INTEGER ID(NUMN,NDE)
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER NFIX, NEQ, IE
*
*     INITIALISATION
*
      NFIX=0
      NEQ=0
*
```

```
*     VALIDATION DU FLAG
*
      IF(FLAG.NE.0.AND.FLAG.NE.1) THEN
        WRITE(IPRINT,1005) FLAG
        STOP
      ENDIF
*
*     ================================================================
*     NUMEROTATION DES EQUATIONS
*     --------------------------
      DO 100 I = 1, NUMN
        DO 100 J = 1, NDE
          IDEQ(I,J) = 0
          IE = ID(I,J)
          IF ( IE.EQ.0 ) THEN
            NEQ = NEQ + 1
            IDEQ(I,J) = NEQ
          ELSEIF ( IE.EQ.1 ) THEN
            IDEQ(I,J) = 0
          ELSE
            NFIX = NFIX + 1
            IDEQ(I,J) = - NFIX
          ENDIF
100   CONTINUE
*     ----------------------------------------
*     VALIDATION DU NOMBRE DE CONDITIONS LIMITES
*     ----------------------------------------
      IF( FLAG.EQ.0) THEN
        NEQT = NEQ
        IF ( NFIX.NE.NFIXB ) THEN
          WRITE(IPRINT,1000) NFIX,NFIXB
          STOP
        ENDIF
      ELSE
        NEQT1 = NEQ
        IF( NFIX.NE.NFIXB1) THEN
          WRITE(IPRINT,1001) NFIX,NFIXB1
          STOP
        ENDIF
      ENDIF
*     ----------------------------------------------------------------
*     MESSAGE D'ERREUR
*     ----------------------------------------------------------------
1000  FORMAT(/,'ERREUR DANS "EQNUM" *** ARRET DU PROGRAMME ***',/
     & 60('=')/'LE NOMBRE DE CONDITIONS LIMITES DANS LE TABLEAU',/
     &         'NE CORRESPOND PAS A LA SOMME DES NFIX',/
     & 60('=')/'  DANS ID : # TOTAL DE COND.= ',I3,/
     &         '  DANS "CONTRL"      # COND.= ',I3)
1001  FORMAT(/,'ERREUR DANS "EQNUM" *** ARRET DU PROGRAMME ***',/
     & 60('=')/'LE NOMBRE DE CONDITIONS LIMITES DANS LE TABLEAU',/
     &         'NE CORRESPOND PAS A LA SOMME DES NFIX1',/
     & 60('=')/'  DANS ID1: # TOTAL DE COND.= ',I3,/
     &         '  DANS "CONTRL"      # COND.= ',I3)
1005  FORMAT(/,'ERREUR DANS "EQNUM" *** ARRET DU PROGRAMME ***',/,
     &         'FLAG = ',I3,' DOIT ETRE 0 OU 1')
*     ----------------------------------------------------------------
      RETURN
      END
*
*     ================================================================
      SUBROUTINE PRTNOD( ID, IDEQ, COORD, FLAG, NUMN, NDE )
*     ================================================================
*
*     FONCTION : IMPRESSION DES TABLEAUX ID,IDEQ ET COORD
*     --------
*
```

```
*     APPEL : CALL PRTNOD ( IA(NID), IA(NIDEQ), IA(NNCOORD), FLAG,
*                           NUMN, NDE )
*     -----
*
*     ARGUMENTS
*     ---------
*
*     ID ................ TABLEAU DES CODES DE CONDITION LIMITE
*     IDEQ ............... TABLEAU DES NUMEROS D'EQUATIONS
*     COORD .............. TABLEAU DES COORDONNEES
*     FLAG ............... 0 => EQUATIONS DE MOUVEMENT
*                          1 => EQUATIONS DE CONFORMATION
*     NUMN ............... NOMBRE DE NOEUDS
*     NDE ................ NOMBRE DE VARIABLES DEPENDANTES
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER MAXVAR
      PARAMETER ( MAXVAR = 4 )
      INTEGER IPRINT
      COMMON / DISKS / IPRINT
      INTEGER IDIM, ITEMP, NCOORD
      COMMON / CONTRL1 / IDIM, ITEMP, NCOORD
      INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
      COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
     &                   NUMNP1, NNPE1, NDEP1
      INTEGER LDOF, KDOF
      COMMON / DOF / LDOF(MAXVAR), KDOF(MAXVAR)
*
*     ARGUMENTS
*     ---------
      INTEGER FLAG, NUMN, NDE
      INTEGER ID(NUMN,NDE), IDEQ(NUMN,NDE)
      REAL*8 COORD(NUMN,NCOORD)
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER IDTEMP(MAXVAR), IDEQT(MAXVAR)
      REAL*8  COORDT(3)
*
*     ================================================================
*
*     VALIDATION DU FLAG
*
      IF(FLAG.NE.0.AND.FLAG.NE.1) THEN
        WRITE(IPRINT,1500) FLAG
        STOP
      ENDIF
      IF(FLAG.EQ.0) THEN
        WRITE(IPRINT,1000)
        DO 100 I = 1,NUMN
          DO 200 J = 1,MAXVAR
            IDTEMP(J) = 0
            IDEQT(J) = 0
            IF (LDOF(J).NE.0) THEN
              IDTEMP(J) = ID(I,LDOF(J))
              IDEQT(J) = IDEQ(I,LDOF(J))
            ENDIF
200       CONTINUE
          DO 300 J = 1,3
            COORDT(J) = 0.0D0
            IF (NCOORD.GE.J) COORDT(J) = COORD(I,J)
300       CONTINUE
          WRITE(IPRINT,1200) I,(IDTEMP(J),J=1,MAXVAR),
     &                         (IDEQT(J),J=1,MAXVAR),
     &                         (COORDT(J),J=1,3)
```

```
100     CONTINUE
      ELSE
        WRITE(IPRINT,1001)
        DO 400 I=1,NUMN
          DO 500 J=1,3
            IDTEMP(J) = ID(I,J)
            IDEQT(J) = IDEQ(I,J)
500       CONTINUE
          DO 600 J=1,3
            COORDT(J) = 0.0D0
            IF(NCOORD.GE.J) COORDT(J) = COORD(I,J)
600       CONTINUE
          WRITE(IPRINT,1201) I,(IDTEMP(J),J=1,3),
     &                         (IDEQT(J),J=1,3),
     &                         (COORDT(J),J=1,3)
400     CONTINUE
      ENDIF
      WRITE(IPRINT,1150)
*     ------------------------------------------------------------
*     FORMAT D'IMPRESSION DES TABLEAUX
*     ------------------------------------------------------------
1000  FORMAT(//'I=====I',2(21('='),'I'),28('='),'I'/'I NOD I',T15,
     &       'ID(I,J)',T29,'I',T36,'IDEQ(I,J)',T51,'I',T60,
     &       'COORDONNEES',T80,'I',/
     &       'I=====I',2(21('='),'I'),28('='),'I'/'I   # I',T12,
     &       'U',T17,'V',T22,'W',T27,'T I',T34,'U',T39,'V',T44,
     &       'W',T49,'T I',T53,'COORD #1',T62,'COORD #2',T71,
     &       'COORD #3 I',/'I=====I',2(21('='),'I'),28('='),'I')
1001  FORMAT(//'I=====I',2(21('='),'I'),28('='),'I'/'I NOD I',T15,
     &       'ID1(I,J)',T29,'I',T36,'IDEQ1(I,J)',T51,'I',T60,
     &       'COORDONNEES',T80,'I',/
     &       'I=====I',2(21('='),'I'),28('='),'I'/'I   # I',T12,
     &       'CXX',T18,'CYY',T24,'CXY  I',T34,'CXX',T40,'CYY',T46,
     &       'CXY  I',T53,'COORD #1',T62,'COORD #2',T71,
     &       'COORD #3 I',/'I=====I',2(21('='),'I'),28('='),'I')
1150  FORMAT('I=====I',2(21('='),'I'),28('='),'I')
1200  FORMAT('I',I4,' I',2(4(I5),' I'),3(F9.5),' I')
1201  FORMAT('I',I4,' I',2(3(I6),'   I'),3(F9.5),' I')
1500  FORMAT(/,'ERREUR DANS "PRTNOD" *** ARRET DU PROGRAMME ***',/,
     &         'FLAG = ',I3,' DOIT ETRE 0 OU 1')
*     ------------------------------------------------------------
      RETURN
      END
*
*     ================================================================
      SUBROUTINE INCONS( CONSTR, ID, IDEQ, FLAG, NUMN, NDE, NFIX )
*     ================================================================
*
*     FONCTION : LECTURE DES CONDITIONS LIMITES NON NULLES
*     --------   GENERE LE TABLEAU CONSTR ( NFIXB )
*
*     APPEL : CALL INCONS( IA(NCONSTR), IA(NID), IA(NIDEQ),
*                          0 OU 1, NUMNP, NDEP, NFIXB )
*     -----
*
*     ARGUMENTS
*     ---------
*
*     CONSTR ........... TABLEAU DES CONDITIONS DE DIRICHLET NON-NULLES
*     ID ............... TABLEAU DES CODES DE CONDITION LIMITE
*     IDEQ ............. TABLEAU DE NUMEROTATION DES EQUATIONS
*     FLAG ............. 0 => EQUATIONS DU MOUVEMENT
*                        1 => EQUATIONS DE CONFORMATION
*     NUMN ............. NOMBRE DE NOEUDS
*     NDE .............. NOMBRE DE VARIABLES DEPENDANTES
*     NFIX ............. NOMBRE DE CONDITIONS DE DIRICHLET NON-NULLES
```

```
*
*     DEFINITION DES VARIABLES LOCALES
*     --------------------------------
*
*     TITRE ............ TITRE DEFINISSANT LA VARIABLE CONSIDEREE
*     NVAL ............. NOMBRE DE CONDITIONS A LIRE POUR UNE VARIABLE
*     VALEUR ........... VALEUR DE LA CONDITION LUE
*     IDEP ............. COMPTEUR ( NDEP)
*     NOEUD ............ NUMERO DU NOEUD
*     IC ............... VARIABLE TEMPORAIRE = - IDEQ ( NOEUD, IDEP )
*
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER MAXVAR
      PARAMETER ( MAXVAR = 4 )
      INTEGER IPRINT
      COMMON / DISKS / IPRINT
      INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
      COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
     &                   NUMNP1, NNPE1, NDEP1
      INTEGER NFIXB, IFIX, NFIXB1, IFIX1
      COMMON / FIX / NFIXB, IFIX(MAXVAR), NFIXB1, IFIX1(3)
      INTEGER LDOF, KDOF
      COMMON / DOF / LDOF(MAXVAR), KDOF(MAXVAR)
*
*     ARGUMENTS
*     ---------
      INTEGER FLAG, NUMN, NDE, NFIX
      INTEGER ID(NUMN,NDE), IDEQ(NUMN,NDE)
      REAL*8 CONSTR(NFIX)
*
*     VARIABLES LOCALES
*     -----------------
      CHARACTER*26 TITRE
      INTEGER NVAL, IDEP, NOEUD, N, IC
      REAL*8 VALEUR
*
*     ================================================================
*
      DO 100 IDEP = 1, NDE
        IF(FLAG.EQ.0) THEN
          NVAL = IFIX(KDOF(IDEP))
        ELSEIF(FLAG.EQ.1) THEN
          NVAL = IFIX1(IDEP)
        ELSE
          WRITE(IPRINT,1400) FLAG
          STOP
       ENDIF
        IF (NVAL.GT.0) THEN
          READ(1,1000) TITRE
          DO 200 I = 1, NVAL
            READ(1,*) NOEUD, VALEUR
*           ---------------------------------------
*           VALIDATION DU CODE DE CONDITION LIMITE
*           ---------------------------------------
            IF (ID(NOEUD,IDEP).NE.2) THEN
              WRITE(IPRINT,1300)
              IF(FLAG.EQ.0) THEN
                WRITE(IPRINT,1200) NOEUD, ID(NOEUD,IDEP)
                STOP
              ELSE
                WRITE(IPRINT,1201) NOEUD, ID(NOEUD,IDEP)
                STOP
              ENDIF
            ENDIF
```

```
*           --------------------
*           STOCKAGE DANS CONSTR
*           --------------------
            IC = -IDEQ(NOEUD,IDEP)
            CONSTR(IC) = VALEUR
200       CONTINUE
        ENDIF
100   CONTINUE
*     ----------------------------------------------------------------
*     FORMAT DE LECTURE ET MESSAGE D'ERREURS
*     ----------------------------------------------------------------
1000  FORMAT(A25)
1100  FORMAT(I5,F11.8)
1200  FORMAT(/,'LE NOEUD #',I4,' NE POSSEDE PAS LE BON CODE',
     &               ' DANS LE TABLEAU ID : ID = ',I4)
1201  FORMAT(/,'LE NOEUD #',I4,' NE POSSEDE PAS LE BON CODE',
     &               ' DANS LE TABLEAU ID1 : ID1 = ',I4)
1300  FORMAT(/,'IL Y A UNE ERREUR DANS LA NUMEROTATION DES',
     &          ' CONDITIONS LIMITES')
1400  FORMAT(/,'ERREUR DANS "INCONS" *** ARRET DU PROGRAMME ***',/,
     &          'FLAG = ',I3,' DOIT ETRE 0 OU 1')
*     ----------------------------------------------------------------
      RETURN
      END
*
*     ================================================================
      SUBROUTINE PRTCON( CONSTR, IDEQ, FLAG, NUMN, NDE, NFIX )
*     ================================================================
*
*     FONCTION : IMPRIME LE TABLEAU CONSTR ( NFIXB )
*     --------
*
*     APPEL : CALL PRTCON ( IA(NCONSTR), IA(NIDEQ), 0 OU 1,
*                           NUMNP, NDEP, NFIXB )
*     -----
*
*     ARGUMENTS
*
*     CONSTR ........... TABLEAU DES CONDITIONS DE DIRICHLET NON-NULLES
*     IDEQ ............. TABLEAU DES NUMEROS D'EQUATIONS
*     FLAG ............. 0 => EQUATIONS DE MOUVEMENT
*                        1 => EQUATIONS DE CONFORMATION
*     NUMN ............. NOMBRE DE NOEUDS
*     NDE .............. NOMBRE DE VARIABLES DEPENDANTES
*     NFIX ............. NOMBRE DE CONDITION DE DIRICHLET NON-HOMOGENES
*
*     DEFINITION DES VARIABLES LOCALES
*     --------------------------------
*
*     NOEUD ............ NUMERO DU NOEUD
*     NFIX ............. COMPTEUR ( NOMBRE DE CONDITIONS LIMITES )
*     NOM .............. NOM DES VARIABLES
*
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER MAXVAR
      PARAMETER ( MAXVAR = 4 )
      INTEGER IPRINT
      COMMON / DISKS / IPRINT
      INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
      COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
     &                   NUMNP1, NNPE1, NDEP1
      INTEGER NFIXB, IFIX, NFIXB1, IFIX1
      COMMON / FIX / NFIXB, IFIX(MAXVAR), NFIXB1, IFIX1(3)
      INTEGER LDOF, KDOF
```

```
      COMMON / DOF / LDOF(MAXVAR), KDOF(MAXVAR)
*
*     ARGUMENTS
*     ---------
      INTEGER FLAG, NUMN, NDE, NFIX
      INTEGER IDEQ(NUMN,NDE)
      REAL*8 CONSTR(NFIX)
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER NOEUD, NFIXCF
      CHARACTER*1 NOM(MAXVAR)
      CHARACTER*3 NOM1(3)
      DATA NOM /'U','V','W','T'/
      DATA NOM1 /'CXX','CYY','CXY'/
*
*     ================================================================
*
      IF(FLAG.EQ.0) THEN
        WRITE(IPRINT,1000)
      ELSEIF(FLAG.EQ.1) THEN
        WRITE(IPRINT,1001)
      ELSE
        WRITE(IPRINT,1500) FLAG
        STOP
      ENDIF
      NFIXCF = 0
      DO 100 I = 1, NUMN
        DO 100 J = 1, NDE
          IF (IDEQ(I,J).LT.0) THEN
            NOEUD = I
            NFIXCF = NFIXCF + 1
            IF(FLAG.EQ.0) THEN
              WRITE(IPRINT,1100) NOEUD, NOM(KDOF(J)), CONSTR(NFIXCF)
            ELSE
              WRITE(IPRINT,1101) NOEUD, NOM1(J),      CONSTR(NFIXCF)
            ENDIF
          ENDIF
100   CONTINUE
      WRITE(IPRINT,1200)
*     ----------------------------------------------------------------
*     FORMAT D'IMPRESSION DU TABLEAU CONSTR
*     ----------------------------------------------------------------
1000  FORMAT(/,T4,'TABLEAU CONSTR(NFIXB)'/30('=')/
     &         'NOEUD',5X,'  NOM',9X,'VALEUR'/30('='))
1001  FORMAT(/,T4,'TABLEAU CONSTR1(NFIXB1)'/30('=')/
     &         'NOEUD',5X,'  NOM',9X,'VALEUR'/30('='))
1100  FORMAT(I5,9X,A1,F15.6)
1101  FORMAT(I5,7X,A3,F15.6)
1200  FORMAT(30('='))
1500  FORMAT(/,'ERREUR DANS "PRTCON" *** ARRET DU PROGRAMME ***',/,
     &         'FLAG = ',I3,' DOIT ETRE 0 OU 1')
*     ----------------------------------------------------------------
      RETURN
      END
*
*     ================================================================
      SUBROUTINE INELEM( CONNEC , LCIEL, DIAG, LMGLOB, IDEQ,
     &                   FLAG, NUMN, NDE, NEQ, MELDO, NNP )
*     ================================================================
*
*     FONCTION : INPUT DES DONNEES SUR LES ELEMENTS
*     --------
*
*     APPEL :  CALL INELEM(IA(NCONNEC), IA(NLCIEL), IA(NDIAG),
```

```
*     -----                  IA(NLMGLOB), IA(NIDEQ), 0 OU 1,
*                            NUMNP, NDEP, NEQT, MELDOF, NNPE)
*
*     ARGUMENTS
*     ---------
*
*     CONNEC ..... . TABLE DE CONNECTIVITE DES ELEMENTS
*     LCIEL ..... PROFIL EN LIGNE DE CIEL DE LA MATRICE GLOBALE
*     DIAG ..... POSITION DES ELEMENTS DE LA DIAGONALE
*     LMGLOB ..... STRUTURE D'ASSEMBLAGE DES ELEMENTS
*     IDEQ ..... TABLEAU DES NUMEROS D'EQUATIONS
*     FLAG ..... 0 => EQUATIONS DU MOUVEMENT
*                1 => EQUATIONS DE CONFORMATION
*     NUMN ..... NOMBRE DE NOEUDS
*     NDE ..... NOMBRE DE VARIABLES DEPENDANTES
*     NEQ ..... NOMBRE TOTAL D'EQUATION
*     MELDO ..... DEGRES DE LIBERTE MAXIMUM D'UN ELEMENT
*     NNP ..... NOMBRE DE NOEUDS PAR ELEMENT
*
*     DEFINITION DES VARIABLES LOCALES
*     --------------------------------
*
*     IGR ..... COMPTEUR (NUMERO DU GROUPE D'ELEMENTS)
*     IELEM ..... COMPTEUR (NUMERO LOCAL DE L'ELEMENT)
*     NELEMG ..... COMPTEUR (NUMERO GLOBAL DE L'ELEMENT)
*     NEL ..... NUMERO LOCAL DE L'ELEMENT LU
*     IC ..... VARIABLE TEMPORAIRE (POINTEUR)
*     CONNECT(MXNNPE) .... TABLE DE CONNECTIVITE TEMPORAIRE DES ELEMENTS
*     LM(MAXDOF) ..... VECTEUR D'ASSEMBLAGE TEMPORAIRE
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER MAXVAR, MXNNPE, MAXDOF
      PARAMETER ( MAXVAR = 4 , MXNNPE = 9, MAXDOF = 36 )
      INTEGER IPRINT
      COMMON / DISKS / IPRINT
      INTEGER IDIM, ITEMP, NCOORD
      COMMON / CONTRL1 / IDIM, ITEMP, NCOORD
      INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
      COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
     &                   NUMNP1, NNPE1, NDEP1
      INTEGER LDOF, KDOF
      COMMON / DOF / LDOF(MAXVAR), KDOF(MAXVAR)
      INTEGER NEQT, MELDOF, ESPMAT
      COMMON / MATDIM / NEQT, MELDOF, ESPMAT
      INTEGER NEQT1, MELDOF1, ESPMAT1
      COMMON / MATDIM1 / NEQT1, MELDOF1, ESPMAT1
      REAL*8 PROP
      INTEGER BLOC
      COMMON / PARAM / PROP(5,4), BLOC(5,8)
*
*     ARGUMENTS
*     ---------
      INTEGER FLAG, NUMN, NDE, NEQ, MELDO, NNP
      INTEGER CONNEC(NELEM,NNP), IDEQ(NUMN,NDE),
     &        LCIEL(NEQ), DIAG(NEQ), LMGLOB(NELEM,MELDO)
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER IGR, IELEM, NELEMG, NEL, IC, ESPMA, NDPIGR
      INTEGER CONNECT(MXNNPE), LM(MAXDOF)
      INTEGER TYPE(5), NELGRP(5), NDP(5), INTGV(5),
     &        INTGP(5), MVISC(5), MCP(5), MCOND(5)
      REAL*8  RHO(5), VISCOS(5), CP(5), COND(5)
      EQUIVALENCE (BLOC(1,1),TYPE(1)),
     &            (BLOC(1,2),NELGRP(1)),
```

```
     &             (BLOC(1,3),NDP(1)),
     &             (BLOC(1,4),INTGV(1)),
     &             (BLOC(1,5),INTGP(1)),
     &             (BLOC(1,6),MVISC(1)),
     &             (BLOC(1,7),MCP(1)),
     &             (BLOC(1,8),MCOND(1)),
     &             (PROP(1,1),RHO(1)),
     &             (PROP(1,2),VISCOS(1)),
     &             (PROP(1,3),CP(1)),
     &             (PROP(1,4),COND(1))
      CHARACTER*75 TITRE,LIGNE
*
*     VALIDATION DU FLAG
*
      IF(FLAG.NE.0.AND.FLAG.NE.1) THEN
        WRITE(IPRINT,1800) FLAG
        STOP
      ENDIF
*
*     ================================================================
*     INITIALISATION
*     --------------
      NELEMG = 0
      DO 50 I = 1, NEQ
        LCIEL(I) = 0
        DIAG(I) = 0
50    CONTINUE
*     ----------------------------
*     BOUCLE SUR LE NOMBRE DE GROUPE
*     ----------------------------
      DO 100 IGR = 1, NGROUP
*       ------------------------
*       INPUT DU BLOC DESCRIPTEUR
*       ------------------------
        IF(FLAG.EQ.0) THEN
          READ(1,1001) TITRE
          READ(1,1001) LIGNE
          READ(1,*) TYPE(IGR), NELGRP(IGR), NDP(IGR), INTGV(IGR),
     &              INTGP(IGR), MVISC(IGR), MCP(IGR), MCOND(IGR)
          NDPIGR = NDP(IGR)
          IF ( ITEMP.NE.0 ) THEN
            READ(1,1001) LIGNE
            READ(1,*) RHO(IGR), VISCOS(IGR), CP(IGR), COND(IGR)
          ELSE
            READ(1,1001) LIGNE
            READ(1,*) RHO(IGR), VISCOS(IGR)
            PROP(IGR,3) = 0.0D0
            PROP(IGR,4) = 0.0D0
          ENDIF
*       ----------------------------
*       VALIDATION DU BLOC DESCRIPTEUR
*       ----------------------------
          IF (TYPE(IGR).GT.3.OR.TYPE(IGR).LT.1) THEN
            WRITE(IPRINT,1400) IGR
            WRITE(IPRINT,1210) TYPE(IGR)
            STOP
          ENDIF
          IF (NELGRP(IGR).GT.NELEM) THEN
            WRITE(IPRINT,1400) IGR
            WRITE(IPRINT,1220) NELGRP(IGR), NELEM
            STOP
          ENDIF
          IF (NDP(IGR).NE.9.AND.NDP(IGR).NE.4) THEN
            WRITE(IPRINT,1400) IGR
            WRITE(IPRINT,1230) NDP(IGR)
            STOP
```

```
            ENDIF
            IF (INTGV(IGR).GT.4.OR.INTGV(IGR).LT.1) THEN
              WRITE(IPRINT,1400) IGR
              WRITE(IPRINT,1240) IGR, INTGV(IGR)
              STOP
            ENDIF
            IF (INTGP(IGR).GT.4.OR.INTGP(IGR).LT.1) THEN
              WRITE(IPRINT,1400) IGR
              WRITE(IPRINT,1250) INTGP(IGR)
              STOP
            ENDIF
            IF (MVISC(IGR).GT.2.OR.MVISC(IGR).LT.0) THEN
              WRITE(IPRINT,1400) IGR
              WRITE(IPRINT,1260) MVISC(IGR)
              STOP
            ENDIF
*           -------------------------
*           VALIDATION DES PROPRIETES
*           -------------------------
            DO 110 I = 1, 4
              IF (PROP(IGR,I).LT.0) THEN
                WRITE(IPRINT,1400) IGR
                WRITE(IPRINT,1270) PROP(IGR,I)
                STOP
              ENDIF
110         CONTINUE
*           ---------------------------------------
*           IMPRESSION DU BLOC DESCRIPTEUR AU FICHIER
*           ---------------------------------------
            WRITE(IPRINT,*)
            WRITE(IPRINT,1001) TITRE
            WRITE(IPRINT,1002)
            WRITE(IPRINT,1000) (BLOC(IGR,I),I=1,8)
            IF (ITEMP.NE.0) THEN
              WRITE(IPRINT,1100) (PROP(IGR,I),I=1,4)
            ELSE
              WRITE(IPRINT,1200) (PROP(IGR,I),I=1,2)
            ENDIF
            WRITE(IPRINT,1002)
          ENDIF
*         -------------------------------------------------
*         LECTURE DE LA TABLE DE CONNECTIVITE DES ELEMENTS
*         -------------------------------------------------
          IF(FLAG.EQ.1) THEN
              NDPIGR = 4
          ENDIF
*
          READ(1,1001) LIGNE
          DO 200 IELEM = 1, NELGRP(IGR)
            DO 300 I = 1, NNP
300         CONNECT(I) = 0
            NELEMG = NELEMG + 1
            READ(1,*) NEL, (CONNECT(I),I=1,NDPIGR)
*           ---------------------------------------
*           VALIDATION DE LA TABLE DE CONNECTIVITE
*           ---------------------------------------
            IF (NEL.GT.NELGRP(IGR)) THEN
              WRITE(IPRINT,1400) IGR
              WRITE(IPRINT,1600) NEL, NELGRP(IGR)
              STOP
            ENDIF
            DO 400 I = 1, NNP
            IF (CONNECT(I).GT.NUMN.OR.CONNECT(I).LT.0) THEN
              WRITE(IPRINT,1400) IGR
              WRITE(IPRINT,1500) CONNECT(I),NUMN
              STOP
```

```
          ELSE
            CONNEC(NELEMG,I) = CONNECT(I)
          ENDIF
400       CONTINUE
*         ------------------------------
*         CALCUL DU VECTEUR D'ASSEMBLAGE
*         ------------------------------
          DO 450 I = 1, MELDO
450         LM(I) = 0
          DO 500 I = 1, NDPIGR
            ND = CONNECT(I)
            DO 500 J = 1, NDE
              IC = I + NDPIGR*(J-1)
              LM(IC) = IDEQ(ND,J)
500       CONTINUE
*         --------------------------------
*         CALCUL DU PROFIL EN LIGNE DE CIEL
*         --------------------------------
          CALL PROFIL(MELDO,NEQ,LM,LCIEL)
*
          DO 600 I = 1, MELDO
600         LMGLOB(NELEMG,I) = LM(I)
200     CONTINUE
100   CONTINUE
*     ------------------------------------------------
*     CALCUL DE LA POSITION DES ELEMENTS DE LA DIAGONALE
*     ------------------------------------------------
      DO 700 I = 1, NEQ
700     DIAG(I) = LCIEL(I)
      CALL GENDIA(NEQ, DIAG, ESPMA, FLAG)
      IF(FLAG.EQ.0) THEN
        WRITE(IPRINT,1700) ESPMAT
      ELSE
        WRITE(IPRINT,1701) ESPMAT1
      ENDIF
*     ---------------------------------------------------------
*     FORMATS DE LECTURE ET MESSAGES D'ERREURS
*     ---------------------------------------------------------
1000  FORMAT(7HTYPE  =,I3,5X,7HNELGRP=,I3,5X,7HNDP   =,I3,5X,7HINTGV =,
     &       I3/,7HINTGP =,I3,5X,7HMVISC =,I3,5X,7HMCP   =,I3,5X,
     &       7HMCOND =,I3)
1001  FORMAT(A75)
1002  FORMAT(70('='))
1200  FORMAT(7HRHO   =,F11.4/,7HVISCOS=,F11.4)
1100  FORMAT(7HRHO   =,F11.4/,7HVISCOS=,F11.4/,7HCP    =,F11.4/,
     &       7HCONDUC=,F11.4)
1210  FORMAT('LA VALEUR DE TYPE DOIT ETRE EGALE A 1 OU 2'/'TYPE =',I3)
1220  FORMAT('LE NOMBRE D''ELEMENTS DANS LE GROUPE EST SUPERIEUR A',
     &       ' NELEM'/'NB. ELEMENTS   = ',I3,/
     &                'MAXIMUM PERMIS =',I3)
1230  FORMAT('LA VALEUR DE NDP DOIT ETRE EGALE A 4 OU 9'/'NDP =',I3)
1240  FORMAT('LA VALEUR DE INTGV DOIT ETRE COMPRISE ENTRE 1 ET 4'/
     &       'INTGV = ',I3)
1250  FORMAT('LA VALEUR DE INTGP DOIT ETRE COMPRISE ENTRE 1 ET 4'/
     &       'INTGP = ',I3)
1260  FORMAT('LA VALEUR DE MVISC DOIT ETRE EGALE A 0,1 OU 2 '/
     &       'MVISC = ',I3)
1270  FORMAT('UNE DES PROPRIETES DU FLUIDE EST NEGATIVE',F10.5)
1300  FORMAT(I5,5X,10(I5))
1400  FORMAT(/,'ERREUR DANS "INELEM" POUR LE GROUPE',I3,
     &         '  *** ARRET DU PROGRAMME ***'/70('='))
1500  FORMAT('LE NUMERO DU NOEUD EST SUPERIEUR AU NOMBRE',
     &       ' PERMIS'/70('=')/'NOEUD LU =',I5/,
     &                        'NUMNP    =',I5)
1600  FORMAT('LE NUMERO DE L''ELEMENT EST SUPERIEUR AU NOMBRE PERMIS'/
     &       70('=')/'NUMERO DE L''ELEMENT  =',I5,/
```

```
     &               'NB ELEMENT DU GROUPE =',I5)
1700  FORMAT(/,'ESPACE MATRICE "A" =',I8,/32('='))
1701  FORMAT(/,'ESPACE MATRICE "A1"=',I8,/32('='))
1800  FORMAT(/,'ERREUR DANS "INELEM" *** ARRET DU PROGRAMME ***',/,
     &         'FLAG = ',I3,' DOIT ETRE 0 OU 1')
*     ------------------------------------------------------------------
      RETURN
      END
*
*     ==================================================================
      SUBROUTINE PROFIL(NBRDLE,NBREQ ,VASELM,HLCIEL)
*     ==================================================================
*
*     FONCTION : FAIT LA MISE A JOUR DE LA LONGUEUR DES BRAS DE LA
*     --------   LIGNE DE CIEL. APPELEE POUR CHAQUE ELEMENT
*
*     APPEL    : CALL PROFIL(NBRDLE,NBREQ ,VASELM,HLCIEL)
*     -----
*
*     ARGUMENTS
*     ---------
*
*     NBRDLE.......... NOMBRE DE DEGRES DE LIBERTE DE L'ELEMENT
*     NBREQ........... NOMBRE D'EQUATIONS DU SYSTEME GLOBAL
*     VASELM(NBRDLE)... VECTEUR D'ASSEMBLAGE DE L'ELEMENT
*     HLCIEL(NBREQ).... LONGUEUR DES BRAS DE LA LIGNE DE CIEL
*
*     ------------------------------------------------------------------
*     NOTES    : 1- HCIEL(NBREQ) DOIT ETRE INITIALISE A ZERO
*                   LE PREMIER APPEL A PROFIL
*                2- PROFIL DOIT ETRE APPELEE POUR CHACUN DES ELEMENTS
*                   DU MAILLAGE
*     ------------------------------------------------------------------
*
*     ARGUMENTS
*     ---------
*
      INTEGER    NBRDLE,NBREQ
      INTEGER    VASELM(NBRDLE)
      INTEGER    HLCIEL(NBREQ )
*
*     LES VARIABLES LOCALES
*     ---------------------
*
      INTEGER    NDLI  ,NDLJ
      INTEGER    EQI   ,EQJ   ,HLOC
*
*     ==================================================================
*     CALCUL DE LA LIGNE DE CIEL.
*     ---------------------------
      DO 2000 NDLI = 1, NBRDLE
         EQI = VASELM(NDLI  )
         IF (EQI.GT.0) THEN
            DO 1000 NDLJ = 1, NBRDLE
               EQJ = VASELM(NDLJ  )
               IF (EQJ.GT.0) THEN
                  IF (EQJ.LT.EQI) THEN
                     HLOC       = EQI - EQJ
                     HLCIEL(EQI) = MAX0(HLOC,HLCIEL(EQI))
                  END IF
               END IF
 1000       CONTINUE
         END IF
 2000 CONTINUE
      RETURN
      END
```

```
*
*     ================================================================
      SUBROUTINE GENDIA(NBREQ, DIAG, ESPMA,FLAG )
*     ================================================================
*
*     FONCTION : CALCUL LE VECTEUR DIAG(NBREQ) DES ADRESSES DES TERMES
*     --------   DIAGONAUX DE LA MATRICE EN LIGNE DE CIEL
*                NON SYMMETRIQUE
*
*     APPEL    : CALL GENDIA ( NBREQ , DIAG  , ESPMA, FLAG)
*     -----
*
*     ARGUMENTS
*     ---------
*
*     NBREQ..... NOMBRE D'EQUATIONS
*     DIAG(NBREQ)..... ENTREE : LONGUEUR DES BRAS DE LA LIGNE DE CIEL
*                      SORTIE : VECTEUR DE LOCALISATION DES TERMES
*                               DIAGONAUX
*     ESPMA ..... NOMBRES DE COEFFICIENTS A STOCKER DANS LA
*                 LIGNE DE CIEL
*     FLAG ..... 0 => EQUATIONS DU MOUVEMENT
*                1 => EQUATIONS DE CONFORMATION
*
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER NEQT, MELDOF, ESPMAT
      COMMON / MATDIM / NEQT, MELDOF, ESPMAT
      INTEGER NEQT1, MELDOF1, ESPMAT1
      COMMON / MATDIM1 / NEQT1, MELDOF1, ESPMAT1
*
*     LES ARGUMENTS
*     -------------
*
      INTEGER    NBREQ, FLAG
      INTEGER    DIAG  (NBREQ),ESPMA
*
*     LES VARIABLES LOCALES
*     ---------------------
*
      INTEGER    NUMEQ ,ISOMME
*
*     ================================================================
*     1> CALCUL DE L'ESPACE NECESSAIRE POUR LE STOCKAGE DE LA MATRICE
*
*        ESPMA = TRIANGLE INF. + DIAGONALE + TRIANGLE SUP.
*        ------------------------------------------------------------
      ISOMME = 0
      DO 1000 NUMEQ = 1, NBREQ
         ISOMME = ISOMME + DIAG (NUMEQ)
 1000 CONTINUE
      ESPMA = 2 * ISOMME + NBREQ
      IF(FLAG.EQ.0) THEN
        ESPMAT = ESPMA
      ELSE
        ESPMAT1 = ESPMA
      ENDIF
*        ---------------------------------
*     2> CALCUL DU VECTEUR DE LOCALISATION
*        ---------------------------------
      DIAG (1) = 1
      DO 1100 NUMEQ = 2, NBREQ
         DIAG(NUMEQ) = DIAG (NUMEQ-1) + 2 * DIAG (NUMEQ) + 1
 1100 CONTINUE
      RETURN
```

```
      END
*
*     ================================================================
      SUBROUTINE PRTTAB( CONNEC, DIAG, LCIEL, FLAG, NNP, NEQ )
*     ================================================================
*
*     FONCTION : IMPRESSION DA LA TABLE DE CONNECTIVITE
*     --------   ET DES VECTEURS PROFIL ET DIAG
*
*     APPEL : CALL ( IA(NCONNEC), IA(NDIAG), IA(NLCIEL), 0 OU 1,
*                    NNPE, NEQT )
*     -----
*
*     ARGUMENTS
*     ---------
*
*     CONNEC ..... TABLE DE CONNECTIVITE DES ELEMENTS
*     DIAG ....... POSITION DES COEFFICIENTS DE LA DIAGONALE
*     LCIEL ...... PROFIL EN LIGNE DE CIEL DE LA MATRICE
*     FLAG ....... 0 => EQUATIONS DU MOUVEMENT
*                  1 => EQUATIONS DE CONFORMATION
*     NNP ........ NOMBRE DE NOEUDS PAR ELEMENT
*     NEQ ........ NOMBRE TOTAL D'EQUATIONS
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER IPRINT
      COMMON / DISKS / IPRINT
      INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
      COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
     &                   NUMNP1, NNPE1, NDEP1
*
*     ARGUMENTS
*     ---------
      INTEGER FLAG, NNP, NEQ
      INTEGER CONNEC(NELEM,NNP), DIAG(NEQ), LCIEL(NEQ)
*
*     VARIABLE LOCALE
*     ---------------
      INTEGER CONNECT(9)
*
*     ================================================================
      WRITE(IPRINT,1000) (I,I=1,9)
      DO 100 I = 1, NELEM
        DO 110 J = 1, 9
          CONNECT(J) = 0
          IF (J.LE.NNP) CONNECT(J) = CONNEC(I,J)
110     CONTINUE
        WRITE(IPRINT,1100) I, (CONNECT(J),J=1,9)
100   CONTINUE
      WRITE(IPRINT,1200)
*      IF(FLAG.EQ.0) THEN
*       WRITE(IPRINT,1300)
*      ELSEIF(FLAG.EQ.1) THEN
*       WRITE(IPRINT,1301)
*      ELSE
*       WRITE(IPRINT,1800) FLAG
*       STOP
*      ENDIF
*     DO 200 I = 1, NEQ
*200     WRITE(IPRINT,1400) I, LCIEL(I), DIAG(I)
*      WRITE(IPRINT,1500)
*     ----------------------------------------------------------------
*     FORMATS D'IMPRESSION
*     ----------------------------------------------------------------
1000  FORMAT(//,T18,'TABLE DE CONNECTIVITE'//'I======I',46('='),'I'/
```

```
     &       'I ELEM I',T17,'NUMEROTATION LOCALE DES NOEUDS',T55,'I'/
     &       'I======I',46('='),'I'/
     &       'I    # I',9(I5),' I'/'I======I',46('='),'I')
1100  FORMAT('I',I5,' I',9(I5),' I')
1200  FORMAT('I======I',46('='),'I')
1300  FORMAT(//,T4,'STRUCTURE EN LIGNE DE CIEL'//'I======I',2(11('='),
     &      'I')/'I EQ.# I',T11,'LCIEL(I) I',T24,'DIAG(I) I'/'I======I',
     &       2(11('='),'I'))
1301  FORMAT(//,T4,'STRUCTURE EN LIGNE DE CIEL'//'I======I',2(11('='),
     &      'I')/'I EQ.# I',T10,'LCIEL1(I) I',T23,'DIAG1(I) I'/
     &      'I======I',2(11('='),'I'))
1400  FORMAT('I',I5,' I',2(I10,' I'))
1500  FORMAT('I======I',2(11('='),'I'))
1800  FORMAT(/,'ERREUR DANS "PRTTAB" *** ARRET DU PROGRAMME ***',/,
     &         'FLAG = ',I3,' DOIT ETRE 0 OU 1')
*     ------------------------------------------------------------------
      RETURN
      END
*
*     ==================================================================
      SUBROUTINE PRTLMG( LMGLOB )
*     ==================================================================
*
*     FONCTION : IMPRESSION DE LA MATRICE GLOBALE D'ASSEMBLAGE
*     --------
*
*     APPEL : CALL ( IA(NLMGLOB) )
*     -----
*
*     ARGUMENTS
*     ---------
*
*     LMGLOB ..... STRUCTURE D'ASSEMBLAGE DES ELEMENTS
*
*     VARIABLES GLOBALES
*     ------------------
      INTEGER MAXVAR
      PARAMETER (MAXVAR = 4)
      INTEGER IPRINT
      COMMON / DISKS / IPRINT
      INTEGER LDOF, KDOF
      COMMON / DOF / LDOF(MAXVAR), KDOF(MAXVAR)
      INTEGER NUMNP, NELEM, NGROUP, NNPE, NDEP, NUMNP1, NNPE1, NDEP1
      COMMON / CONTRL2 / NUMNP, NELEM, NGROUP, NNPE, NDEP,
     &                   NUMNP1, NNPE1, NDEP1
      INTEGER NEQT, MELDOF, ESPMAT
      COMMON / MATDIM / NEQT, MELDOF, ESPMAT
*
*     ARGUMENTS
*     ---------
      INTEGER LMGLOB(NELEM, MELDOF)
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER LMTEMP(16)
*
*     ==================================================================
*
      IF (NNPE.LE.4) THEN
      WRITE(IPRINT,1000)
      DO 100 I = 1, NELEM
        DO 150 L = 1, 16
150       LMTEMP(L) = 0
*         ------------------------------------------------
*         AFFICHAGE DES RESULTATS SOUS FORME NON COMPACTE
*         ------------------------------------------------
```

```
          DO 200 J = 1 , NNPE
            DO 300 K = 1, NDEP
              IF (KDOF(K).NE.0) THEN
                IC = J + (KDOF(K)-1)*NNPE
                LMTEMP(IC) = LMGLOB(I,J + NNPE*(K-1))
              ENDIF
300         CONTINUE
200       CONTINUE
        WRITE(IPRINT,1100) I, (LMTEMP(J),J=1,16)
100   CONTINUE
      WRITE(IPRINT,1200)
      ENDIF
*     ----------------------------------------------------------------
*     FORMAT D'IMPRESSION
*     ----------------------------------------------------------------
1000  FORMAT(/,'STRUCTURE D''ASSEMBLAGE'/'I======I',4(17('='),'I')/
     &        'I ELEM I',4(3X,'1',3X,'2',3X,'3',3X,'4',' I')/
     &        'I======I',4(17('='),'I')/
     &        'I    # I',4(3X,'U'),' I', 4(3X,'V'),' I',4(3X,'W'),' I',
     &         4(3X,'T'),' I',/'I======I',4(17('='),'I'))
1100  FORMAT('I',I5,' I',4(4(I4),' I'))
1200  FORMAT('I======I',4(17('='),'I'))
*     ----------------------------------------------------------------
      RETURN
      END
*
*     ================================================================
      SUBROUTINE MODSOL
*     ================================================================
*
*  FONCTION : SOLUTION DU PROBLEME
*
*
*  VARIABLES LOCALES
*
*  ITER              COMPTEUR (NOMBRE D'ITERATION)
*  IGR               COMPTEUR (NOMBRE DE GROUPE)
*  MAXITER           NOMBRE MAXIMUM D'ITERATIONS (VITESSES)
*  MAXITER1          NOMBRE MAXIMUM D'ITERATIONS (CONFORMATION)
*  MAXITGL           NOMBRE MAXIMUM D'ITERATIONS VISCOELASTIQUES
*  IELEMG            COMPTEUR (NUMERO D'ELEMENT GLOBAL)
*
*  VARIABLES GLOBALES
*  ------------------
      IMPLICIT NONE
      INTEGER MBLANK
      PARAMETER (MBLANK= 2500000)
      INTEGER IA
      COMMON /BLANK/ IA(MBLANK)
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
      COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                 NUMNP1,NNPE1,NDEP1
      INTEGER BLOC
      REAL*8 PROP
      COMMON /PARAM/ PROP(5,4),BLOC(5,8)
      INTEGER TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      COMMON /BLOC/ TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      REAL*8 RHO,VISCOS,CP,COND
      COMMON/ PROP/ RHO,VISCOS,CP,COND
      INTEGER NEQT,MELDOF,ESPMAT
      COMMON /MATDIM/ NEQT,MELDOF,ESPMAT
      INTEGER NEQT1,MELDOF1,ESPMAT1
      COMMON /MATDIM1/ NEQT1,MELDOF1,ESPMAT1
      INTEGER MAXITER, MAXITER1, MAXITGL
```

```
      COMMON /ITER/ MAXITER, MAXITER1, MAXITGL
      REAL*8 DE
      COMMON /DEBORAH/ DE
      INTEGER NEWTON
      COMMON /PREMIERE/ NEWTON
*
*  VARIABLES LOCALES
*  -----------------
      INTEGER ITER,ITER1,ITERGL,IGR,NA,NX,NB,NDX,NDIAG,IELEMG,
     &        NA1,NX1,NB1,NDX1,NDIAG1,NNCOORD1,NCONNEC1,NLMGLOB1,
     &        NNCOORD,NCONNEC,NLMGLOB,ADDTBL,LOCTBL
      INTEGER TYPE1, NDP1, INTGC, FLAG, COMPTEUR
      REAL*8  N2X,N2R,N2DX,NMX,NMR,NMDX,NORML2,NORMAX,ECART,NORMS
      LOGICAL CONVRG
*
*  ALLOCATION DES TABLEAUX DES 2 SYSTEMES GLOBAUX DANS IA
*
      NA       = ADDTBL('A',     ESPMAT,  'REAL', 'MODSOL')
      NA1      = ADDTBL('A1',    ESPMAT1, 'REAL', 'MODSOL')
      NX       = ADDTBL('X',     NEQT,    'REAL', 'MODSOL')
      NX1      = ADDTBL('X1',    NEQT1,   'REAL', 'MODSOL')
      NB       = ADDTBL('B',     NEQT,    'REAL', 'MODSOL')
      NB1      = ADDTBL('B1',    NEQT1,   'REAL', 'MODSOL')
      NDX      = ADDTBL('DX',    NEQT,    'REAL', 'MODSOL')
      NDX1     = ADDTBL('DX1',   NEQT1,   'REAL', 'MODSOL')
      NDIAG    = LOCTBL('DIAG',    'MODSOL')
      NDIAG1   = LOCTBL('DIAG1',   'MODSOL')
      NNCOORD  = LOCTBL('COORD',   'MODSOL')
      NNCOORD1 = LOCTBL('COORD1',  'MODSOL')
      NCONNEC  = LOCTBL('CONNEC',  'MODSOL')
      NCONNEC1 = LOCTBL('CONNEC1', 'MODSOL')
      NLMGLOB  = LOCTBL('LMGLOB',  'MODSOL')
      NLMGLOB1 = LOCTBL('LMGLOB1', 'MODSOL')
      CALL LISTBL('MODSOL')
*
*  CALCUL DE LA SOLUTION PAR ITERATIONS
*
      CALL INITX(IA(NX),NEQT,IA(NX1),NEQT1)
      DO 700 ITERGL=1,MAXITGL
        CONVRG = .TRUE.
*
*  ITERATIONS EQUATIONS DU MOUVEMENT
*
        DO 100 ITER=1,MAXITER
        IELEMG=0
*
*  INITIALISATION DES MESURES DE LA CONVERGENCE
*
        IF(ITER.EQ.1) THEN
          ECART=0.0D0
          NORMS=0.0D0
          COMPTEUR=0
        ENDIF
*
*  INITIALISATION DU SYSTEME GLOBAL (A*X=B)
*
        CALL INITG(IA(NA),IA(NB),IA(NDX), ESPMAT, NEQT)
        DO 200 IGR=1,NGROUP
*
*  RECUPERATION DU BLOC DESCRIPTEUR ET DES PROPRIETES
*
          TYPE    =BLOC(IGR,1)
          NELGRP  =BLOC(IGR,2)
          NDP     =BLOC(IGR,3)
          INTGV   =BLOC(IGR,4)
          INTGP   =BLOC(IGR,5)
```

```
            MVISC   =BLOC(IGR,6)
            MCP     =BLOC(IGR,7)
            MCOND   =BLOC(IGR,8)
            RHO     =PROP(IGR,1)
            VISCOS  =PROP(IGR,2)
            CP      =PROP(IGR,3)
            COND    =PROP(IGR,4)
*
*  CALCUL DES COEFFICIENTS DE LA MATRICE GLOBALE (A)
*
            CALL STOKES(IELEMG,IA(NCONNEC),IA(NNCOORD),IA(NLMGLOB),
     &                  IA(NLMGLOB1),ITER,ITERGL,ECART,NORMS,CONVRG,
     &                  COMPTEUR)
 200      CONTINUE
*          goto 101
*
*  FACTORISATION LU ET RESOLUTION DU SYSTEME D'EQUATIONS
*
          CALL FACTLU(NEQT,ESPMAT,IA(NDIAG),IA(NA))
          CALL RESOUD(NEQT,ESPMAT,IA(NDIAG),IA(NA),IA(NB),IA(NDX))
*
*  CALCUL DES NORMES EUCLEDIENNES ET MAXIMALES
*
          N2X     =NORML2(IA(NX),NEQT)
          N2DX    =NORML2(IA(NDX),NEQT)
          N2R     =NORML2(IA(NB),NEQT)
          NMX     =NORMAX(IA(NX),NEQT)
          NMDX    =NORMAX(IA(NDX),NEQT)
          NMR     =NORMAX(IA(NB),NEQT)
*
*  MISE A JOUR DU VECTEUR SOLUTION
*
        IF(NEWTON.EQ.0.AND.ITERGL.EQ.1) THEN
          FLAG=1
        ELSE
          FLAG=0
        ENDIF
          CALL MISAJR(IA(NX),IA(NDX),NEQT,FLAG)
*
*  IMPRESSION DES NORMES ET TEST DE CONVERGENCE (MOUVEMENT)
*
          IF(ITER.EQ.1) WRITE(IPRINT,1050)
          IF(ITER.EQ.1) WRITE(IPRINT,1000)
          WRITE(IPRINT,1100) ITER,N2X,N2DX,N2R,NMX,NMDX,NMR
*
            IF(NMDX.LT.1.0E-05.AND.NMR.LT.1.0E-05.AND.
     &         N2DX.LT.1.0E-04.AND.N2R.LT.1.0E-04) THEN
               WRITE(IPRINT,1200)
               GO TO 101
            ENDIF
 100    CONTINUE
        WRITE(IPRINT,1200)
        WRITE(IPRINT,1400) MAXITER
        STOP
 101    CONTINUE
*
*        goto 123
*
*  ITERATIONS EQUATIONS DE CONFORMATION
*
        DO 500 ITER1=1,MAXITER1
          IELEMG=0
*
*  INITIALISATION DU SYSTEME GLOBAL (A1*X1=B1)
*
          CALL INITG(IA(NA1),IA(NB1),IA(NDX1),ESPMAT1,NEQT1)
```

```
          DO 600 IGR=1,NGROUP
            TYPE1 = 1
            NDP1  = 4
            INTGC = 2
*
*  CALCUL DES COEFFICIENTS DE LA MATRICES GLOBALE (A1)
*
            CALL CONF(IELEMG,IA(NCONNEC),IA(NCONNEC1),IA(NNCOORD),
     &                IA(NNCOORD1),IA(NLMGLOB),IA(NLMGLOB1),NDP1,
     &                ITERGL)
 600      CONTINUE
*
*  FACTORISATION LU ET RESOLUTION DU SYSTEME D'EQUATIONS
*
          CALL FACTLU(NEQT1,ESPMAT1,IA(NDIAG1),IA(NA1))
          CALL RESOUD(NEQT1,ESPMAT1,IA(NDIAG1),IA(NA1),
     &                IA(NB1),IA(NDX1))
*
*  CALCUL DES NORMES
*
          N2X    =NORML2(IA(NX1),NEQT1)
          N2DX   =NORML2(IA(NDX1),NEQT1)
          N2R    =NORML2(IA(NB1),NEQT1)
          NMX    =NORMAX(IA(NX1),NEQT1)
          NMDX   =NORMAX(IA(NDX1),NEQT1)
          NMR    =NORMAX(IA(NB1),NEQT1)
*
*  MISE A JOUR DU VECTEUR SOLUTION
*
          CALL MISAJR(IA(NX1),IA(NDX1),NEQT1,0)
*
*  IMPRESSION DES NORMES ET TEST DE CONVERGENCE (CONFORMATION)
*
          IF(ITER1.EQ.1) WRITE(IPRINT,1051)
          IF(ITER1.EQ.1) WRITE(IPRINT,1000)
          WRITE(IPRINT,1100) ITER1,N2X,N2DX,N2R,NMX,NMDX,NMR
          IF(NMDX.LT.1.0E-05.AND.NMR.LT.1.0E-05.AND.
     &       N2DX.LT.1.0E-04.AND.N2R.LT.1.0E-04) THEN
             WRITE(IPRINT,1200)
             GO TO 501
          ENDIF
 500    CONTINUE
        WRITE(IPRINT,1200)
        WRITE(IPRINT,1401) MAXITER1
        STOP
 501    CONTINUE
*
* 123    continue
*
*  TEST DE CONVERGENCE GLOBALE (CONTRAINTES)
*
        IF(CONVRG.AND.(NEWTON.EQ.1.OR.ITERGL.GE.2)) THEN
          WRITE(IPRINT,1300) ITERGL
          RETURN
        ELSEIF(NEWTON.NE.0.OR.ITERGL.NE.1) THEN
          WRITE(IPRINT,1500) ITERGL, ECART, NORMS
          WRITE(IPRINT,1501) COMPTEUR
        ENDIF
 700  CONTINUE
      WRITE(IPRINT,1402) MAXITGL
*
 1000 FORMAT(/,T32,'CALCUL DES NORMES'//'I======I',6(11('='),'I')/
     &       'I ITER I',T18,'NORME EUCLEDIENNE',T44,'I',T55,
     &       'VALEUR MAXIMALE',T80,'I'/'I======I',6(11('='),'I')/
     &       'I      I',T18,'X I',T29,'DX I',T42,'R I',T54,'X I',
     &       T65,'DX I',T78,'R I'/'I======I',6(11('='),'I'))
```

```
 1050 FORMAT(/,T30,'EQUATIONS DU MOUVEMENT')
 1051 FORMAT(/,T28,'EQUATIONS DE CONFORMATION')
 1100 FORMAT('I',I5,' I',2(2(G10.4,' I'),G10.2,' I'))
 1200 FORMAT('I======I',6(11('='),'I'))
 1300 FORMAT(/,I3,' ITERATIONS VISCOELASTIQUES: *** CONVERGENCE ***'/)
 1400 FORMAT(/,I4,' ITERATIONS EFFECTUEES (MOUVEMENT)',/,T5,
     &         'CONVERGENCE NON ATTEINTE *** ARRET DU PROGRAMME ***')
 1401 FORMAT(/,I3,' ITERATIONS EFFECTUEES (CONFORMATION)',/,T5,
     &         'CONVERGENCE NON ATTEINTE *** ARRET DU PROGRAMME ***')
 1402 FORMAT(/,I3,' ITERATIONS VISCOELASTIQUES EFFECTUEES',/,T5,
     &         'CONVERGENCE NON ATTEINTE'/)
 1500 FORMAT(/,' ITERATION #',I3,' SOMME ECARTS SIJ = ',E10.4,2X,
     &                          ' NORME INFINIE SIJ = ',E10.4)
 1501 FORMAT(16X,'CONVERGENCE NON SATISFAITE (SIJ) EN',1X,I5,
     &           ' ENDROITS')
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE STOKES(IELEMG,CONNEC,COORD,LMGLOB,LMGLOB1,
     &                  ITER,ITERGL,ECART,NORMS,CONVRG,COMPTEUR)
*     ================================================================
*
*  FONCTION : CALCUL DES SYSTEMES ELEMENTAIRES POUR LES EQUATIONS
*             DE STOKES : FORMULATION DE PENALISATION PAR INTEGRATION
*             REDUITE
*
*  ARGUMENTS
*    IELEMG       COMPTEUR (NUMEROTATION GLOBALE DES ELEMENTS)
*    CONNEC       TABLE DE CONNECTIVITE DES ELEMENTS
*    COORD        TABLE DE COORDONNEES DES NOEUDS
*    LMGLOB       TABLE DES VECTEURS D'ASSEMBLAGE DES ELEMENTS (MOUVEMENT)
*    LMGLOB1      TABLE DES VECTEURS D'ASSEMBLAGE DES ELEMENTS (CONFORMATION)
*    ITER         NUMERO DE L'ITERATION (MOUVEMENT)
*    ITERGL       NUMERO DE L'ITERATION VISCOELASTIQUE
*
*  VARIABLES LOCALES
*
*    COORDE       TABLE DES COORDONNES GLOBALE DES ELEMENTS
*    NINTGU       NOMBRE DE POINTS DE GAUSS POUR LA MATRICE DE DIFFUSION
*    NINTGP       NB DE POINTS DE GAUSS POUR LA MATRICE DE PENALISATION
*    NINTGC       NB DE POINTS DE GAUSS POUR LA MATRICE DE CONFORMATION
*    IELEM        COMPTEUR (NOMBRE D'ELEMENTS)
*
*  VARIABLES GLOBALES
*  ------------------
      IMPLICIT NONE
      INTEGER MBLANK
      PARAMETER (MBLANK= 2500000)
      INTEGER IA
      COMMON /BLANK/ IA(MBLANK)
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      INTEGER IDIM,ITEMP,NCOORD
      COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
      INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
      COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                 NUMNP1,NNPE1,NDEP1
      INTEGER LDOF,KDOF
      COMMON /DOF/ LDOF(4),KDOF(4)
      INTEGER NFIXB,IFIX,NFIXB1,IFIX1
      COMMON /FIX/ NFIXB,IFIX(4),NFIXB1,IFIX1(3)
      INTEGER TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      COMMON /BLOC/ TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      REAL*8 RHO,VISCOS,CP,COND
```

```
      COMMON /PROP/ RHO,VISCOS,CP,COND
      INTEGER NEQT,MELDOF,ESPMAT
      COMMON /MATDIM/ NEQT,MELDOF,ESPMAT
      INTEGER NEQT1,MELDOF1,ESPMAT1
      COMMON /MATDIM1/ NEQT1,MELDOF1,ESPMAT1
      REAL*8 EPS
      COMMON/EPS/ EPS
*
*  VARIABLES LOCALES
*  -----------------
      INTEGER NINTGU,NINTGP,NINTGC,NMAT,NBEL,NXGU,NXGP,NXGC,NWU,NWP,
     &        NWC,NSHU,NSHC,NDLU,NDLC,NDGU,NJAU,NSHP,NDLP,NDGP,NJAP,
     &        IELEM,NX,NX1,NB,NCONSTR,NCONSTR1,NDIAG,NA,NOEUD,NETA,
     &        NRGU,NRGP,NSXX,NSYY,NSXY,NSXXIM1,NSYYIM1,NSXYIM1,
     &        TYPE1,NDP1,INTGC,
     &        ADDTBL,LOCTBL
      REAL*8  COORDE(9,2)
      INTEGER LM(36),LM1(12),I,J
*
*  ARGUMENTS
*  ---------
      INTEGER IELEMG,CONNEC(NELEM,NDP),LMGLOB(NELEM,MELDOF),
     &        LMGLOB1(NELEM,MELDOF1),ITER,ITERGL,COMPTEUR
      REAL*8  COORD(NUMNP,NCOORD),ECART,NORMS
      LOGICAL CONVRG
*
*  PROPRIETES ET CARACTERISTIQUES DE L'ELEMENT
*
      TYPE1 = 1
      NDP1  = 4
      INTGC = 2
*
*  CALCUL DU NOMBRE DE POINT D'INTEGRATION EN VITESSE, EN PRESSION
*  ET EN CONFORMATION
*
      NINTGU=INTGV**NCOORD
      NINTGP=INTGP**NCOORD
      NINTGC=INTGC**NCOORD
*
*  ALLOCATION DES TABLEAUX DANS IA
*
      NMAT=ADDTBL('MATELM',MELDOF*MELDOF,'REAL','STOKES')
      NBEL=ADDTBL('BELM',MELDOF,'REAL','STOKES')
      NXGU=ADDTBL('XGAUSU',NINTGU*NCOORD,'REAL','STOKES')
      NXGP=ADDTBL('XGAUSP',NINTGP*NCOORD,'REAL','STOKES')
      NXGC=ADDTBL('XGAUSC',NINTGC*NCOORD,'REAL','STOKES')
      NWU=ADDTBL('WU',NINTGU,'REAL','STOKES')
      NWP=ADDTBL('WP',NINTGP,'REAL','STOKES')
      NWC=ADDTBL('WC',NINTGC,'REAL','STOKES')
      NSHU=ADDTBL('SHAPEU',NINTGU*NDP,'REAL','STOKES')
      NDLU=ADDTBL('DSLOCU',NINTGU*NDP*NCOORD,'REAL','STOKES')
      NDGU=ADDTBL('DSGLBU',NINTGU*NDP*NCOORD,'REAL','STOKES')
      NJAU=ADDTBL('DJACU',NINTGU,'REAL','STOKES')
      NSHP=ADDTBL('SHAPEP',NINTGP*NDP,'REAL','STOKES')
      NDLP=ADDTBL('DSLOCP',NINTGP*NDP*NCOORD,'REAL','STOKES')
      NDGP=ADDTBL('DSGLBP',NINTGP*NDP*NCOORD,'REAL','STOKES')
      NJAP=ADDTBL('DJACP',NINTGP,'REAL','STOKES')
      NSHC=ADDTBL('SHAPEC',NINTGC*NDP1,'REAL','STOKES')
      NDLC=ADDTBL('DSLOCC',NINTGC*NDP1*NCOORD,'REAL','STOKES')
      NETA=ADDTBL('ETA',NINTGU,'REAL','STOKES')
      NRGU=ADDTBL('RAYONU',NINTGU,'REAL','STOKES')
      NRGP=ADDTBL('RAYONP',NINTGP,'REAL','STOKES')
      NSXX=ADDTBL('SXX',NINTGU,'REAL','STOKES')
      NSYY=ADDTBL('SYY',NINTGU,'REAL','STOKES')
      NSXY=ADDTBL('SXY',NINTGU,'REAL','STOKES')
      NSXXIM1=ADDTBL('SXXIM1',NELEM*NINTGU,'REAL','STOKES')
```

```
      NSYYIM1=ADDTBL('SYYIM1',NELEM*NINTGU,'REAL','STOKES')
      NSXYIM1=ADDTBL('SXYIM1',NELEM*NINTGU,'REAL','STOKES')
      NCONSTR=LOCTBL('CONSTR','STOKES')
      NCONSTR1=LOCTBL('CONSTR1','STOKES')
      NDIAG=LOCTBL('DIAG','STOKES')
      NA=LOCTBL('A','STOKES')
      NB=LOCTBL('B','STOKES')
      NX=LOCTBL('X','STOKES')
      NX1=LOCTBL('X1','STOKES')
*
*  CALCUL DES POIDS ET POINTS D'INTEGRATION
*
      CALL GAUSS(INTGV,NCOORD,NINTGU,IA(NXGU),IA(NWU))
      CALL FBASE(NINTGU,NDP,NCOORD,IA(NXGU),IA(NSHU),
     &           IA(NDLU),TYPE)
      CALL GAUSS(INTGP,NCOORD,NINTGP,IA(NXGP),IA(NWP))
      CALL FBASE(NINTGP,NDP,NCOORD,IA(NXGP),IA(NSHP),
     &           IA(NDLP),TYPE)
      CALL GAUSS(INTGC,NCOORD,NINTGC,IA(NXGC),IA(NWC))
      CALL FBASE(NINTGC,NDP1,NCOORD,IA(NXGC),IA(NSHC),
     &           IA(NDLC),TYPE1)
*
*  BOUCLE SUR LES ELEMENTS DU GROUPE
*
       DO 100 IELEM=1,NELGRP
         IELEMG=IELEMG+1
*
*  CALCUL DES COORDONNEES REELLES DE L'ELEMENT
*
         DO 200 I=1,NDP
           NOEUD=CONNEC(IELEMG,I)
           DO 200 J=1,NCOORD
             COORDE(I,J)=COORD(NOEUD,J)
 200     CONTINUE
*
*  CALCUL DES DERIVEES GLOBALES AUX POINTS D'INTEGRATION
*
         CALL INITE(IA(NMAT),IA(NBEL),MELDOF)
         CALL DRGLB(NINTGU,NDP,NCOORD,COORDE,
     &              IA(NDLU),IA(NDGU),IA(NJAU))
         CALL DRGLB(NINTGP,NDP,NCOORD,COORDE,
     &              IA(NDLP),IA(NDGP),IA(NJAP))
         CALL RAYON(NINTGU,COORDE,IA(NXGU),IA(NRGU))
         CALL RAYON(NINTGP,COORDE,IA(NXGP),IA(NRGP))
*
*  RECUPERATION DU VECTEUR D'ASSEMBLAGE
*
         DO 300 I=1,MELDOF
 300       LM(I)=LMGLOB(IELEMG,I)
         DO 301 I=1,MELDOF1
 301       LM1(I)=LMGLOB1(IELEMG,I)
*
*  CALCUL DE LA VISCOSITE (NEWTONNIEN GENERALISE)
*** POUR LE MODELE VISCOELASTIQUE ETA(G)=VISCOS ***
*** IL FAUT IMPOSER DANS LE FICHIER DE DONNEES MVISC=0 ***
*
         CALL RHEO(IA(NETA),IA(NDGU),IA(NSHU),LM,NINTGU,ITER,
     &             NEQT,NFIXB,MELDOF,IA(NX),IA(NCONSTR),IA(NRGU))
*
*  CALCUL DE LA MATRICE ELEMENTAIRE ET DU MEMBRE DE DROITE
*
         CALL MATDIF(IA(NSHU),IA(NDGU),IA(NJAU),IA(NETA),
     &               IA(NWU),IA(NMAT),NINTGU,
     &               NDP,MELDOF,IA(NRGU))
         CALL MATPEN(IA(NSHP),IA(NDGP),IA(NJAP),EPS,
     &               IA(NWP),IA(NMAT),NINTGP,
```

```
     &                NDP,MELDOF,IA(NRGP))
        CALL SIGMA(ITERGL,IA(NSXX),IA(NSYY),IA(NSXY),NINTGU,NINTGC,
     &             NDP,NDP1,LM,LM1,IA(NX),IA(NX1),IA(NCONSTR),
     &             IA(NCONSTR1),IA(NSHC),IA(NDGU),IA(NWC))
        CALL MEMBD(IA(NBEL),MELDOF,IA(NSXX),IA(NSYY),IA(NSXY),
     &             IA(NDGU),NINTGU,NDP,IA(NJAU),IA(NWU))
*
*  COMPARAISON DU TENSEURS DES CONTRAINTES VISCOELASTIQUES DE
*  L'ITERATION COURANTE AVEC L'ITERATION PRECEDENTE (CONFORMATION)
*
          CALL CONVERGE(ITERGL,NELEM,IELEMG,NINTGU,IA(NSXX),IA(NSYY),
     &                  IA(NSXY),IA(NSXXIM1),IA(NSYYIM1),IA(NSXYIM1),
     &                  ECART,NORMS,CONVRG,COMPTEUR)
*
*  ASSEMBLAGE DES CONDITIONS LIMITES
*
          CALL CNDLIM(NEQT,NFIXB,MELDOF,IA(NX),IA(NCONSTR),
     &                LM,IA(NMAT),IA(NBEL),IA(NB))
*
*  ASSEMBLAGE DU SYSTEME ELEMENTAIRE DANS LA MATRICE GLOBALE
*
          CALL ASSEMB(ESPMAT,NEQT,IA(NDIAG),IA(NA),
     &                LM,MELDOF,IA(NMAT))
 100  CONTINUE
*
*  PURGE DES TABLEAUX DU SYSTEME ELEMENTAIRE
*
       CALL RMVTBL('SXYIM1')
       CALL RMVTBL('SYYIM1')
       CALL RMVTBL('SXXIM1')
       CALL RMVTBL('SXY')
       CALL RMVTBL('SYY')
       CALL RMVTBL('SXX')
       CALL RMVTBL('RAYONP')
       CALL RMVTBL('RAYONU')
       CALL RMVTBL('ETA')
       CALL RMVTBL('DSLOCC')
       CALL RMVTBL('SHAPEC')
       CALL RMVTBL('DJACP')
       CALL RMVTBL('DSGLBP')
       CALL RMVTBL('DSLOCP')
       CALL RMVTBL('SHAPEP')
       CALL RMVTBL('DJACU')
       CALL RMVTBL('DSGLBU')
       CALL RMVTBL('DSLOCU')
       CALL RMVTBL('SHAPEU')
       CALL RMVTBL('WC')
       CALL RMVTBL('WP')
       CALL RMVTBL('WU')
       CALL RMVTBL('XGAUSC')
       CALL RMVTBL('XGAUSP')
       CALL RMVTBL('XGAUSU')
       CALL RMVTBL('BELM')
       CALL RMVTBL('MATELM')
       RETURN
       END
*
*
*      ================================================================
       SUBROUTINE CONF(IELEMG,CONNEC,CONNEC1,COORD,COORD1,LMGLOB,LMGLOB1,
     &                 NDP1,ITERGL)
*      ================================================================
*
*  FONCTION : CALCUL DES SYSTEMES ELEMENTAIRES POUR LES EQUATIONS
*             DE CONFORMATION
*
```

```
*  ARGUMENTS
*    IELEMG       COMPTEUR (NUMEROTATION GLOBALE DES ELEMENTS)
*    CONNEC       TABLE DE CONNECTIVITE DES ELEMENTS (MOUVEMENT)
*    CONNEC1      TABLE DE CONNECTIVITE DES ELEMENTS (CONFORMATION)
*    COORD        TABLE DE COORDONNEES DES NOEUDS (MOUVEMENT)
*    COORD1       TABLE DE COORDONNEES DES NOEUDS (CONFORMATION)
*    LMGLOB       TABLE DES VECTEURS D'ASSEMBLAGE DES ELEMENTS (MOUVEMENT)
*    LMGLOB1      TABLE DES VECTEURS D'ASSEMBLAGE DES ELEMENTS (CONFORMATION)
*    NDP1         NOMBRE DE FONCTION D'INTERPOLATION PAR ELEMENT
*    ITERGL       NUMERO DE L'ITERATION VISCOELASTIQUE
*
*  VARIABLES LOCALES
*
*    COORDE       TABLE DES COORDONNES GLOBALES DES ELEMENTS (MOUVEMENT)
*    COORDE1      TABLE DES COORDONNES GLOBALES DES ELEMENTS (CONFORMATION)
*    NINTGU       NOMBRE DE POINTS DE GAUSS POUR LA MATRICE DE DIFFUSION
*    NINTGP       NB DE POINTS DE GAUSS POUR LA MATRICE DE PENALISATION
*    NINTGC       NB DE POINTS DE GAUSS POUR LA MATRICE DE CONFORMATION
*    IELEM        COMPTEUR (NOMBRE D'ELEMENTS)
*
*  VARIABLES GLOBALES
*  ------------------
      IMPLICIT NONE
      INTEGER MBLANK
      PARAMETER (MBLANK= 2500000)
      INTEGER IA
      COMMON /BLANK/ IA(MBLANK)
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      INTEGER IDIM,ITEMP,NCOORD
      COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
      INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
      COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                 NUMNP1,NNPE1,NDEP1
      INTEGER LDOF,KDOF
      COMMON /DOF/ LDOF(4),KDOF(4)
      INTEGER NFIXB,IFIX,NFIXB1,IFIX1
      COMMON /FIX/ NFIXB,IFIX(4),NFIXB1,IFIX1(3)
      INTEGER TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      COMMON /BLOC/ TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      REAL*8 RHO,VISCOS,CP,COND
      COMMON /PROP/ RHO,VISCOS,CP,COND
      INTEGER NEQT,MELDOF,ESPMAT
      COMMON /MATDIM/ NEQT,MELDOF,ESPMAT
      INTEGER NEQT1,MELDOF1,ESPMAT1
      COMMON /MATDIM1/ NEQT1,MELDOF1,ESPMAT1
      REAL*8 EPS
      COMMON/EPS/ EPS
*
*  VARIABLES LOCALES
*  -----------------
      INTEGER NINTGU,NINTGC,NMAT1,NBEL1,NXGU,NXGC,NWU,
     &        NWC,NSHU,NSHC,NDLU,NDLC,NDGU,NDGC,NJAU,NJAC,
     &        IELEM,NX,NX1,NB1,NCONSTR,NCONSTR1,NDIAG1,NA1,NOEUD,
     &        TYPE1,INTGC,
     &        ADDTBL,LOCTBL
      REAL*8  COORDE(9,2),COORDE1(4,2)
      INTEGER LM(36),LM1(12),I,J
*
*  ARGUMENTS
*  ---------
      INTEGER IELEMG,NDP1,ITERGL
      INTEGER CONNEC(NELEM,NDP),CONNEC1(NELEM,NDP1),
     &        LMGLOB(NELEM,MELDOF),LMGLOB1(NELEM,MELDOF1)
      REAL*8 COORD(NUMNP,NCOORD),COORD1(NUMNP1,NCOORD)
*
```

```
*  PROPRIETES ET CARACTERISTIQUES DE L'ELEMENT
*
      TYPE1 = 1
*     NDP1  = 4
      INTGC = 2
*
*  CALCUL DU NOMBRE DE POINT D'INTEGRATION EN VITESSE, EN PRESSION
*  ET EN CONFORMATION
*
      NINTGU=INTGV**NCOORD
      NINTGC=INTGC**NCOORD
*
*  ALLOCATION DES TABLEAUX DANS IA
*
      NMAT1=ADDTBL('MATELM1',MELDOF1*MELDOF1,'REAL','CONF')
      NBEL1=ADDTBL('BELM1',MELDOF1,'REAL','CONF')
      NXGU=ADDTBL('XGAUSU',NINTGU*NCOORD,'REAL','CONF')
      NXGC=ADDTBL('XGAUSC',NINTGC*NCOORD,'REAL','CONF')
      NWU=ADDTBL('WU',NINTGU,'REAL','CONF')
      NWC=ADDTBL('WC',NINTGC,'REAL','CONF')
      NSHU=ADDTBL('SHAPEU',NINTGU*NDP,'REAL','CONF')
      NDLU=ADDTBL('DSLOCU',NINTGU*NDP*NCOORD,'REAL','CONF')
      NDGU=ADDTBL('DSGLBU',NINTGU*NDP*NCOORD,'REAL','CONF')
      NJAU=ADDTBL('DJACU',NINTGU,'REAL','CONF')
      NSHC=ADDTBL('SHAPEC',NINTGC*NDP1,'REAL','CONF')
      NDLC=ADDTBL('DSLOCC',NINTGC*NDP1*NCOORD,'REAL','CONF')
      NDGC=ADDTBL('DSGLBC',NINTGC*NDP1*NCOORD,'REAL','CONF')
      NJAC=ADDTBL('DJACC',NINTGC,'REAL','CONF')
      NCONSTR=LOCTBL('CONSTR','CONF')
      NCONSTR1=LOCTBL('CONSTR1','CONF')
      NDIAG1=LOCTBL('DIAG1','CONF')
      NA1=LOCTBL('A1','CONF')
      NB1=LOCTBL('B1','CONF')
      NX=LOCTBL('X','CONF')
      NX1=LOCTBL('X1','CONF')
*
*  CALCUL DES POIDS ET POINTS D'INTEGRATION
*
      CALL GAUSS(INTGV,NCOORD,NINTGU,IA(NXGU),IA(NWU))
      CALL FBASE(NINTGU,NDP,NCOORD,IA(NXGU),IA(NSHU),
     &           IA(NDLU),TYPE)
      CALL GAUSS(INTGC,NCOORD,NINTGC,IA(NXGC),IA(NWC))
      CALL FBASE(NINTGC,NDP1,NCOORD,IA(NXGC),IA(NSHC),
     &           IA(NDLC),TYPE1)
*
*  BOUCLE SUR LES ELEMENTS DU GROUPE
*
       DO 100 IELEM=1,NELGRP
         IELEMG=IELEMG+1
*
*  CALCUL DES COORDONNEES REELLES DE L'ELEMENT
*
         DO 200 I=1,NDP
           NOEUD=CONNEC(IELEMG,I)
           DO 200 J=1,NCOORD
             COORDE(I,J)=COORD(NOEUD,J)
 200     CONTINUE
       DO 201 I=1,NDP1
         NOEUD=CONNEC1(IELEMG,I)
         DO 201 J=1,NCOORD
           COORDE1(I,J)=COORD1(NOEUD,J)
 201     CONTINUE
*
*  CALCUL DES DERIVEES GLOBALES AUX POINTS D'INTEGRATION
*
         CALL INITE(IA(NMAT1),IA(NBEL1),MELDOF1)
```

```
          CALL DRGLB(NINTGU,NDP,NCOORD,COORDE,
     &               IA(NDLU),IA(NDGU),IA(NJAU))
          CALL DRGLB(NINTGC,NDP1,NCOORD,COORDE1,
     &               IA(NDLC),IA(NDGC),IA(NJAC))
*   RECUPERATION DU VECTEUR D'ASSEMBLAGE
*
          DO 300 I=1,MELDOF
 300        LM(I)=LMGLOB(IELEMG,I)
        DO 301 I=1,MELDOF1
 301        LM1(I)=LMGLOB1(IELEMG,I)
*
*   CALCUL DE LA MATRICE ELEMENTAIRE ET DU MEMBRE DE DROITE
*
        CALL VISCO(IA(NSHU),IA(NSHC),IA(NJAC),IA(NDGU),IA(NDGC),
     &             IA(NWU),IA(NWC),IA(NMAT1),NINTGU,NINTGC,NDP,NDP1,
     &             LM,LM1,IA(NX),IA(NX1),IA(NCONSTR),IA(NCONSTR1),
     &             COORDE)
        CALL MEMBD1(IA(NBEL1),NINTGC,IA(NSHC),NDP,NDP1,IA(NJAC),IA(NWC),
     &             LM,LM1,IA(NX),IA(NX1),IA(NCONSTR),IA(NCONSTR1),
     &             COORDE1,ITERGL)
*
*   ASSEMBLAGE DES CONDITIONS LIMITES
*
          CALL CNDLIM(NEQT1,NFIXB1,MELDOF1,IA(NX1),IA(NCONSTR1),
     &               LM1,IA(NMAT1),IA(NBEL1),IA(NB1))
*
*   ASSEMBLAGE DU SYSTEME ELEMENTAIRE DANS LA MATRICE GLOBALE
*
          CALL ASSEMB(ESPMAT1,NEQT1,IA(NDIAG1),IA(NA1),
     &               LM1,MELDOF1,IA(NMAT1))
 100  CONTINUE
*
*   PURGE DES TABLEAUX DU SYSTEME ELEMENTAIRE
*
        CALL RMVTBL('DJACC')
        CALL RMVTBL('DSGLBC')
        CALL RMVTBL('DSLOCC')
        CALL RMVTBL('SHAPEC')
        CALL RMVTBL('DJACU')
        CALL RMVTBL('DSGLBU')
        CALL RMVTBL('DSLOCU')
        CALL RMVTBL('SHAPEU')
        CALL RMVTBL('WC')
        CALL RMVTBL('WU')
        CALL RMVTBL('XGAUSC')
        CALL RMVTBL('XGAUSU')
        CALL RMVTBL('BELM1')
        CALL RMVTBL('MATELM1')
        RETURN
        END
*
*
*       ================================================================
        SUBROUTINE INITX(X,NBREQ,X1,NBREQ1)
*       ================================================================
*
*   FONCTION : ESTIME INITIAL DE LA SOLUTION (A UN NOMBRE DE
*              DEBORAH PLUS FAIBLE => FICHIER estime)
*
        INTEGER NEWTON
        COMMON / PREMIERE / NEWTON
        INTEGER NBREQ,NBREQ1
        REAL*8 X(NBREQ),X1(NBREQ1)
        CHARACTER*75 LIGNE
        IF(NEWTON.EQ.1) READ(3,1000) LIGNE
        DO 100 I=1,NBREQ
```

```
        IF(NEWTON.EQ.1) READ(3,*) X(I)
        IF(NEWTON.EQ.0) X(I)=0.0D0
 100  CONTINUE
      DO 200 I=1,NBREQ1
        IF(NEWTON.EQ.1) READ(3,*) X1(I)
        IF(NEWTON.EQ.0) X1(I)=0.0D0
 200  CONTINUE
 1000 FORMAT(A75)
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE INITE(MATELM,BELM,MELDOF)
*     ================================================================
*
*  FONCTION : INITIALISE LE SYSTEME ELEMENTAIRE
*
      INTEGER MELDOF
      REAL*8 MATELM(MELDOF,MELDOF),BELM(MELDOF)
      DO 100 I=1,MELDOF
        BELM(I)=0.0D0
        DO 100 J=1,MELDOF
          MATELM(I,J)=0.0D0
 100  CONTINUE
      RETURN
      END
*
*     ================================================================
      SUBROUTINE INITG(A,B,DX,ESPMAT,NEQT)
*     ================================================================
*
*  FONCTION : INITIALISE LE SYSTEME GLOBAL
*
      INTEGER ESPMAT, NEQT
      REAL*8 A(ESPMAT),  B(NEQT),  DX(NEQT)
      DO 100 I=1,ESPMAT
 100    A(I)=0.0D0
      DO 200 I=1,NEQT
        B(I)=0.0D0
        DX(I)=0.0D0
 200  CONTINUE
      RETURN
      END
*
*     ================================================================
      DOUBLE PRECISION FUNCTION NORMAX(VECTEUR,LENGTH)
*     ================================================================
*
*  FONCTION : TROUVER LA PLUS GRANDE VALEUR DANS UN VECTEUR
*
*  ARGUMENTS :
*    VECTEUR      VECTEUR
*    LENGTH       NOMBRE DE VALEURS DANS LE VECTEUR
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER LENGTH,I
      REAL*8 VECTEUR(LENGTH)
*
*  ON BOUCLE SUR LES VALEURS NE CONSERVANT QUE LA PLUS GRANDE
*
      NORMAX=0.0D00
      DO 998 I=1,LENGTH
 998    IF (DABS(VECTEUR(I)).GT.NORMAX) NORMAX=DABS(VECTEUR(I))
      RETURN
      END
```

```
*
*
*      ================================================================
       DOUBLE PRECISION FUNCTION NORML2(VECTEUR,LENGTH)
*      ================================================================
*
*  FONCTION : CALCULER LA NORME EUCLEDIENNE D'UN VECTEUR
*
* ARGUMENTS :
*   VECTEUR    VECTEUR
*   LENGTH     LONGUEUR DU VECTEUR
*
       IMPLICIT CHARACTER (A-Z)
       INTEGER LENGTH,I
       REAL*8 VECTEUR(LENGTH)
*
       NORML2=0.0D00
       DO 10 I=1,LENGTH
         NORML2=NORML2+VECTEUR(I)**2
 10    CONTINUE
       NORML2=DSQRT(NORML2)
       RETURN
       END
*
*
*      ================================================================
       SUBROUTINE CNDLIM(NBREQ,NBRCNT,NBRDLE,X,CONTNT,
      &                  VASELM,MATELM,BELM,BGLB)
*      ================================================================
*
*  FONCTION : IMPOSER LES CONDITIONS LIMITES SUR LA MATRICE
*
       IMPLICIT CHARACTER (A-Z)
       INTEGER NBREQ,NBRCNT,NBRDLE
       INTEGER VASELM(NBRDLE)
       INTEGER I,J,NUMI,NUMJ
       REAL*8 RESELM,X(NBREQ),CONTNT(NBRCNT)
       REAL*8 BELM(NBRDLE),BGLB(NBREQ)
       REAL*8 MATELM(NBRDLE,NBRDLE)
*
*  MISE A JOUR DU RESIDU GLOBAL
*
       DO 400 I=1,NBRDLE
         NUMI=VASELM(I)
         IF (NUMI.GT.0) THEN
           RESELM=-BELM(I)
           DO 401 J=1,NBRDLE
             NUMJ=VASELM(J)
             IF (NUMJ.GT.0) THEN
               RESELM=RESELM+MATELM(I,J)*X(NUMJ)
             ELSE IF(NUMJ.LT.0) THEN
               RESELM=RESELM+MATELM(I,J)*CONTNT(-NUMJ)
             ENDIF
 401       CONTINUE
           BGLB(NUMI)=BGLB(NUMI)-RESELM
         ENDIF
 400   CONTINUE
*
       RETURN
       END
*
*
*      ================================================================
       SUBROUTINE ASSEMB(ESPMAT,NBREQ,DIAG,MATGLB,VASELM,
      &                  NBRDLE,MATELM)
*      ================================================================
```

```
*
*  FONCTION : ASSEMBLAGE D'UNE MATRICE ELEMENTAIRE DANS LA MATRICE
*             GLOBALE NON SYMMETRIQUE STOCKEE EN LIGNE DE CIEL.
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER ESPMAT,NBREQ,NBRDLE
      INTEGER VASELM(NBRDLE),DIAG(NBREQ)
      INTEGER I,J,IJ,NUMI,NUMJ
      INTEGER ID,IL,IU,LONGI,IMIN
      INTEGER JD,JL,JU,LONGJ,JMIN
      REAL*8 MATELM(NBRDLE,NBRDLE),MATGLB(ESPMAT)
*
*  BOUCLE SUR LES DEGRES DE LIBERTE EN I ET EN J
*
      DO 500 I=1,NBRDLE
        NUMI=VASELM(I)
*
*  1> LE DEGRE DE LIBERTE EST UNE INCONNUE
*     ON ASSEMBLE LA LIGNE I
*
        IF (NUMI.GT.0) THEN
          DO 510 J=1,NBRDLE
            NUMJ=VASELM(J)
*
*  2> LE DEGRE DE LIBERTE EST UNE INCONNUE
*     ON ASSEMBLE LA COLONNE J
*
            IF (NUMJ.GT.0) THEN
*
*  ADRESSAGE DU COEFFICIENT DE LA MATRICE ELEMENTAIRE
*  DANS LA MATRICE GLOBALE
*
              IF (NUMI.EQ.NUMJ) THEN
                IJ=DIAG(NUMI)
              ELSE IF (NUMI.GT.NUMJ) THEN
                  ID=DIAG(NUMI)
                  IL=DIAG(NUMI-1)+1
                  IU=(ID+IL)/2
                  LONGI=IU-IL
                  IMIN=NUMI-LONGI
                  IJ=IL+NUMJ-IMIN
               ELSE IF (NUMI.LT.NUMJ) THEN
                    JD=DIAG(NUMJ)
                    JL=DIAG(NUMJ-1)+1
                    JU=(JD+JL)/2
                    LONGJ=JU-JL
                    JMIN=NUMJ-LONGJ
                    IJ=JU+NUMI-JMIN
                    ENDIF
                    MATGLB(IJ)=MATGLB(IJ)+MATELM(I,J)
              ENDIF
 510        CONTINUE
          ENDIF
 500    CONTINUE
*
      RETURN
      END
*
*
*     ==========================================================
      SUBROUTINE FACTLU(NBREQ,ESPMAT,DIAG,LU)
*     ==========================================================
*
*  FONCTION : FACTORISATION LU VARIANTE DE DOOLITTLE D'UNE MATRICE
*             NON SYMMETRIQUE A PROFIL SYMMETRIQUE, STOCKEE EN LIGNE
*             DE CIEL
```

```
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      INTEGER NBREQ,ESPMAT
      INTEGER DIAG(NBREQ)
      INTEGER I,J,IJ
      INTEGER ID,IL,IU,LONGI,IMIN
      INTEGER JD,JL,JU,LONGJ
      INTEGER DEBI,DEBJ,LONG
      REAL*8 EPSLU,LU(ESPMAT)
      REAL*8 PRDSCL
*
      EPSLU=1.D-08
*
*  BOUCLE SUR LES EQUATIONS
*
      DO 600 I=2,NBREQ
        ID=DIAG(I)
        IL=DIAG(I-1)+1
        IU=(ID+IL)/2
        LONGI=IU-IL
        IMIN=I-LONGI
*
*  CAS POUR J=IMIN
*
        IF(IMIN.NE.I) THEN
          LU(IL)=LU(IL)/LU(DIAG(IMIN))
        ENDIF
*
*  CALCUL DE L(I,J) ET U(J,I)
*
        DO 610 J=IMIN+1,I-1
          JD=DIAG(J)
          JL=DIAG(J-1)+1
          JU=(JD+JL)/2
          LONGJ=JU-JL
          LONG=MIN0(J-IMIN,LONGJ)
*
*  CALCUL DE L(I,J)
*
          IJ=IL+J-IMIN
          DEBI=IJ-LONG
          DEBJ=JD-LONG
          LU(IJ)=(LU(IJ)-PRDSCL(LU(DEBI),LU(DEBJ),LONG))/LU(JD)
*
*  CALCUL DE U(J,I)
*
          IJ=IU+J-IMIN
          DEBI=IJ-LONG
          DEBJ=JU-LONG
          LU(IJ)=LU(IJ)-PRDSCL(LU(DEBI),LU(DEBJ),LONG)
 610    CONTINUE
*
*  CALCUL DE U(I,I)
*
        LU(ID)=LU(ID)-PRDSCL(LU(IL),LU(IU),LONGI)
        IF (DABS( LU(ID)).LE.EPSLU) THEN
          WRITE(IPRINT,5000) LU(ID),I
          STOP
        ENDIF
 600  CONTINUE
      RETURN
*
*  FORMATS
*
```

```
 5000 FORMAT(' FACTLU - PIVOT = ',1PD16.8,' A L" EQUATION NO.',I6)
*
      END
*
*
*     ================================================================
      SUBROUTINE RESOUD(NBREQ,ESPMAT,DIAG,LU,B,X)
*     ================================================================
*
*  FONCTION : RESOLUTION D'UN SYSTEME MATRICIEL NON SYMETRIQUE
*             STOCKE EN LIGNE DE CIEL
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER NBREQ,ESPMAT
      INTEGER DIAG(NBREQ)
      INTEGER I,J
      INTEGER ID,IL,IU,LONGI,IMIN
      INTEGER JD,JL,JU,LONGJ,JMIN
      REAL*8 XJ,X(NBREQ),B(NBREQ),LU(ESPMAT)
      REAL*8 PRDSCL
*
*  DESCENTE TRIANGULAIRE
*
      X(1)=B(1)
*
      DO 700 I=2,NBREQ
        ID=DIAG(I)
        IL=DIAG(I-1)+1
        IU=(ID+IL)/2
        LONGI=IU-IL
        IMIN=I-LONGI
        X(I)=B(I)-PRDSCL(LU(IL),X(IMIN),LONGI)
 700  CONTINUE
*
*  REMONTEE TRIANGULAIRE
*
      DO 710 J = NBREQ, 2 , -1
        JD=DIAG(J)
        JL=DIAG(J-1)+1
        JU=(JD+JL)/2
        LONGJ=JU-JL
        JMIN=J-LONGJ
        XJ=X(J)/LU(DIAG(J))
        CALL SUBSTI(X(JMIN),LU(JU),XJ,J-JMIN)
        X(J)=XJ
 710  CONTINUE
*
      X(1)=X(1)/LU(DIAG(1))
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE SUBSTI(X,U,XJ,LENGTH)
*     ================================================================
*
*  FONCTION : EFFECTUE L'OPERATION VECTORIELLE X=X-XJ*U
*             POUR LA SUBSTITUTION A REBOURS EN LIGNE DE CIEL
*
      INTEGER I,LENGTH
      REAL*8 X(LENGTH),U(LENGTH),XJ
      DO 100 I=1,LENGTH
        X(I)=X(I)-XJ*U(I)
 100  CONTINUE
      RETURN
      END
```

```
*
*
*     ================================================================
      DOUBLE PRECISION FUNCTION PRDSCL (X,Y,LONG)
*     ================================================================
*
*  FONCTION : PRODUIT SCALAIRE DE DEUX VECTEURS DE LONGUEUR "LONG"
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER I,LONG
      REAL*8 SOMME,X(LONG),Y(LONG)
*
*  LE PRODUIT SCALAIRE
*
      SOMME = 0.0D00
      IF(LONG.NE.0) THEN
        DO 850 I=1,LONG
          SOMME=SOMME+X(I)*Y(I)
 850    CONTINUE
      ENDIF
      PRDSCL=SOMME
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE GAUSS(INT,NCOORD,NPGAUS,XGAUSS,WGAUSS)
*     ================================================================
*
*  FONCTION : CALCULE LES POIDS ET LA POSITION DES POINTS D'INTEGRATION
*             POUR UN ELEMENT QUADRILATAIRE
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      LOGICAL ERREUR,ETAT(3)
      INTEGER I,J,K,POSI,INT,NCOORD,NPGAUS
      REAL*8 XG(5),PG(5),XGAUSS(NPGAUS,NCOORD),WGAUSS(NPGAUS)
*
*  VERIFICATION DE LA CORRESPONDANCE ENTRE LES ARGUMENTS
*
      ETAT(1)=.FALSE.
      ETAT(2)=.FALSE.
      ETAT(3)=.FALSE.
      IF ((INT.LT.1).OR.(INT.GT.5)) THEN
        ETAT(1)=.TRUE.
      ENDIF
      IF ((NCOORD.LT.1).OR.(NCOORD.GT.3)) THEN
        ETAT(2)=.TRUE.
      ENDIF
      IF (NPGAUS.NE.(INT**NCOORD)) THEN
        ETAT(3)=.TRUE.
      ENDIF
      ERREUR=ETAT(1).OR.ETAT(2).OR.ETAT(3)
      IF (ERREUR) THEN
        WRITE(IPRINT,9000)
        IF (ETAT(1)) THEN
          WRITE(IPRINT,9001) INT
        ENDIF
        IF (ETAT(2)) THEN
          WRITE(IPRINT,9002) NCOORD
        ENDIF
        IF (ETAT(3)) THEN
          WRITE(IPRINT,9003) NPGAUS,INT,NCOORD
        ENDIF
       STOP
```

```
      END IF
*
*  CALCUL DU POIDS ET DE LA POSITION SELON LE # DE POINTS D'INTEGRATION
*
      DO 10 I=1,5
        XG(I)=0.0D00
        PG(I)=0.0D00
 10   CONTINUE
      IF (INT.EQ.1) THEN
        XG(1)=0.0D00
        PG(1)=2.0D00
      ELSE
        IF (INT.EQ.2) THEN
          XG(1)=-1.0D00/DSQRT(3.0D00)
          XG(2)=-XG(1)
          PG(1)=1.0D00
          PG(2)=1.0D00
        ELSE
          IF (INT.EQ.3) THEN
            XG(1)=-DSQRT(3.0D00/5.0D00)
            XG(2)=0.0D00
            XG(3)=-XG(1)
            PG(1)=5.0D00/9.0D00
            PG(2)=8.0D00/9.0D00
            PG(3)=PG(1)
          ELSE
            IF (INT.EQ.4) THEN
              XG(1)=-DSQRT((3.0D0+2.0*DSQRT(6.0D0/5.0D0))/7.0D0)
              XG(2)=-DSQRT((3.0D0-2.0*DSQRT(6.0D0/5.0D0))/7.0D0)
              XG(3)=-XG(2)
              XG(4)=-XG(1)
              PG(1)=0.5D0-1.0D0/(6.0D0*DSQRT(6.0D0/5.0D0))
              PG(2)=0.5D0+1.0D0/(6.0D0*DSQRT(6.0D0/5.0D0))
              PG(3)=PG(2)
              PG(4)=PG(1)
            ELSE
              IF (INT.EQ.5) THEN
                XG(1)=-DSQRT(5.0D0+4.0D0*DSQRT(5.0D0/14.0D0))/3.0D0
                XG(2)=-DSQRT(5.0D0-4.0D0*DSQRT(5.0D0/14.0D0))/3.0D0
                XG(3)= 0.0D0
                XG(4)=-XG(2)
                XG(5)=-XG(1)
               PG(1)=161.0D0/450.0D0-13.0D0/(180.D0*DSQRT(5.0D0/14.0D0))
               PG(2)=161.0D0/450.0D0+13.0D0/(180.D0*DSQRT(5.0D0/14.0D0))
                PG(3)=128.0D0/225.0D0
                PG(4)=PG(2)
                PG(5)=PG(1)
              ENDIF
            ENDIF
          ENDIF
        ENDIF
      ENDIF
*
*  CALCUL DU POIDS ET DE LA POSITION SELON LES DIMENSIONS DE L'ELEMENT
*
      POSI=1
      IF (NCOORD.EQ.1) THEN
        DO 900 I=1,INT
        WGAUSS(POSI)=PG(I)
        XGAUSS(POSI,1)=XG(I)
        POSI=POSI+1
 900    CONTINUE
      ELSE
        IF(NCOORD.EQ.2) THEN
          DO 902 J=1,INT
            DO 903 I=1,INT
```

```
            WGAUSS(POSI)=PG(I)*PG(J)
            XGAUSS(POSI,1)=XG(I)
            XGAUSS(POSI,2)=XG(J)
            POSI=POSI+1
 903        CONTINUE
 902      CONTINUE
        ELSE
          IF (NCOORD.EQ.3) THEN
            DO 904 K=1,INT
              DO 905 J=1,INT
                DO 906 I=1,INT
                WGAUSS(POSI)=PG(I)*PG(J)*PG(K)
                XGAUSS(POSI,1)=XG(I)
                XGAUSS(POSI,2)=XG(J)
                XGAUSS(POSI,3)=XG(K)
                POSI=POSI+1
 906            CONTINUE
 905          CONTINUE
 904        CONTINUE
          ENDIF
        ENDIF
      ENDIF
      RETURN
*
*  FORMATS
*
 9000 FORMAT('ERREUR DANS LA SOUS ROUTINE GAUSS')
 9001 FORMAT('LE NOMBRE DE POINTS D''INTEGRATION = ',I3)
 9002 FORMAT('LE NOMBRE DE COORDONNEES = ',I3)
 9003 FORMAT('LE # POIDS = ',I7,' INT = ',I3,' A LA NCOORD = ',I3)
      END
*
*     ================================================================
      SUBROUTINE FBASE(NPGAUS,NDP,NCOORD,XGAUSS,SHAPE,DSLOC,TYPE)
*     ================================================================
*
*  FONCTION : CALCULE LES VALEURS DE LA FONCTION D'INTERPOLATION ET
*             DE SES DERIVEES AUX POINTS D'INTEGRATION LOCAUX
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER I,N,G
      REAL*8 KSI,ETA
      INTEGER NPGAUS,NDP,NCOORD,TYPE
      REAL*8 XGAUSS(NPGAUS,NCOORD),SHAPE(NPGAUS,NDP)
      REAL*8 DSLOC(NPGAUS,NDP,NCOORD)
*
*  ELEMENT 2D QUADRILATAIRE A 4 NOEUDS
*
      IF (TYPE.EQ.1) THEN
        DO 910 G=1,NPGAUS
          KSI=XGAUSS(G,1)
          ETA=XGAUSS(G,2)
*
*  VALEURS DES FONCTIONS N1 A N4
*
          SHAPE(G,1)=(1.0D00-KSI)*(1.00D00-ETA)/4.0D00
          SHAPE(G,2)=(1.0D00+KSI)*(1.00D00-ETA)/4.00D00
          SHAPE(G,3)=(1.0D00+KSI)*(1.00D00+ETA)/4.00D00
          SHAPE(G,4)=(1.0D00-KSI)*(1.00D00+ETA)/4.00D00
*
*  VALEURS DES DERIVEES DES FONCTIONS D(N1 A N4)/D(KSI ET ETA)
*
          DSLOC(G,1,1)=(ETA-1.00D00)/4.0D00
          DSLOC(G,2,1)=(1.00D00-ETA)/4.00D00
          DSLOC(G,3,1)=(1.00D00+ETA)/4.00D00
          DSLOC(G,4,1)=(-1.00D00-ETA)/4.00D00
```

```
          DSLOC(G,1,2)=(KSI-1.0D00)/4.00D00
          DSLOC(G,2,2)=(-1.0D00-KSI)/4.00D00
          DSLOC(G,3,2)=(1.0D00+KSI)/4.00D00
          DSLOC(G,4,2)=(1.0D00-KSI)/4.00D00
 910    CONTINUE
      ENDIF
*
*  ELEMENT 2D QUADRILATAIRE A 9 NOEUDS : 3/COTES + CENTRE
*
      IF (TYPE.EQ.2) THEN
        DO 911 G=1,NPGAUS
          KSI=XGAUSS(G,1)
          ETA=XGAUSS(G,2)
*
*  VALEURS DES FONCTIONS N1 A N9
*
          SHAPE(G,1)=((KSI**2)-KSI)*((ETA**2)-ETA)/4.0D00
          SHAPE(G,2)=((KSI**2)+KSI)*((ETA**2)-ETA)/4.0D00
          SHAPE(G,3)=((KSI**2)+KSI)*((ETA**2)+ETA)/4.0D00
          SHAPE(G,4)=((KSI**2)-KSI)*((ETA**2)+ETA)/4.0D00
          SHAPE(G,5)=(1.0D00-(KSI**2))*((ETA**2)-ETA)/2.00D00
          SHAPE(G,6)=((KSI**2)+KSI)*(1.00D00-(ETA**2))/2.00D00
          SHAPE(G,7)=(1.0D00-(KSI**2))*((ETA**2)+ETA)/2.00D00
          SHAPE(G,8)=((KSI**2)-KSI)*(1.00D00-(ETA**2))/2.00D00
          SHAPE(G,9)=(1.0D00-(KSI**2))*(1.00D00-(ETA**2))
*
*  VALEURS DES DERIVEES DES FONCTIONS D(N1 A N9)/D(KSI ET ETA)
*
          DSLOC(G,1,1)=((2.0D0*KSI)-1.0D0)*((ETA**2)-ETA)/4.0D0
          DSLOC(G,2,1)=((2.0D0*KSI)+1.0D0)*((ETA**2)-ETA)/4.0D0
          DSLOC(G,3,1)=((2.0D0*KSI)+1.0D0)*((ETA**2)+ETA)/4.0D0
          DSLOC(G,4,1)=((2.0D0*KSI)-1.0D0)*((ETA**2)+ETA)/4.0D0
          DSLOC(G,5,1)=(-2.0D0*KSI)*((ETA**2)-ETA)/2.0D0
          DSLOC(G,6,1)=((2.0D0*KSI)+1.0D0)*(1.0D0-(ETA**2))/2.0D0
          DSLOC(G,7,1)=(-2.0D0*KSI)*((ETA**2)+ETA)/2.0D0
          DSLOC(G,8,1)=((2.0D0*KSI)-1.0D0)*(1.0D0-(ETA**2))/2.0D0
          DSLOC(G,9,1)=(-2.0D0*KSI)*(1.0D0-(ETA**2))
          DSLOC(G,1,2)=((KSI**2)-KSI)*((2.0D0*ETA)-1.0D0)/4.0D0
          DSLOC(G,2,2)=((KSI**2)+KSI)*((2.0D0*ETA)-1.0D0)/4.0D0
          DSLOC(G,3,2)=((KSI**2)+KSI)*((2.0D0*ETA)+1.0D0)/4.0D0
          DSLOC(G,4,2)=((KSI**2)-KSI)*((2.0D0*ETA)+1.0D0)/4.0D0
          DSLOC(G,5,2)=(1.0D0-(KSI**2))*((2.0D0*ETA)-1.0D0)/2.0D0
          DSLOC(G,6,2)=((KSI**2)+KSI)*(-2.0D0*ETA)/2.0D0
          DSLOC(G,7,2)=(1.0D0-(KSI**2))*((2.0D0*ETA)+1.0D0)/2.0D0
          DSLOC(G,8,2)=((KSI**2)-KSI)*(-2.0D0*ETA)/2.0D0
          DSLOC(G,9,2)=(1.0D0-(KSI**2))*(-2.0D0*ETA)
 911    CONTINUE
      ENDIF
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE DRGLB(NPGAUS,NDP,NCOORD,COORDE,DSLOC,DSGLB,DJAC)
*     ================================================================
*
*  FONCTION : CALCULE LES VALEURS DES DERIVEES DANS LES COORDONNEES GLOBALES
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      INTEGER I,J,G
      INTEGER NPGAUS,NDP,NCOORD
      REAL*8 J1,J2,J3,J4
      REAL*8 DSLOC(NPGAUS,NDP,NCOORD),DSGLB(NPGAUS,NDP,NCOORD)
      REAL*8 DJAC(NPGAUS),COORDE(NDP,2)
```

```
*
*  CALCUL DES DERIVEES GLOBALES EN 2 D
*
      IF (NCOORD.EQ.2) THEN
        DO 920 G=1,NPGAUS
          J1=0.0D0
          J2=0.0D0
          J3=0.0D0
          J4=0.0D0
          DO 20 I=1,NDP
            J1=J1+DSLOC(G,I,1)*COORDE(I,1)
            J2=J2+DSLOC(G,I,1)*COORDE(I,2)
            J3=J3+DSLOC(G,I,2)*COORDE(I,1)
            J4=J4+DSLOC(G,I,2)*COORDE(I,2)
 20       CONTINUE
          DJAC(G)=(J1*J4)-(J3*J2)
          DO 921 I=1,NDP
            DSGLB(G,I,1)=(J4*DSLOC(G,I,1)-J2*DSLOC(G,I,2))/DJAC(G)
            DSGLB(G,I,2)=(-J3*DSLOC(G,I,1)+J1*DSLOC(G,I,2))/DJAC(G)
921       CONTINUE
920     CONTINUE
      ELSE
        WRITE(IPRINT,1001) NCOORD
        STOP
      ENDIF
      RETURN
1001  FORMAT(/ ' SOUS-ROUTINE DRGLB, NCOORD=',I3,/,
     &         ' CAS PARTICULIER DOIT ETRE 2 ,***ARRET DU PROGRAMME***')
      END
*
*
*     ================================================================
      SUBROUTINE RAYON(NPGAUS,COORDE,XGAUSS,R)
*     ================================================================
*
*  FONCTION : CALCUL LES VALEURS DE L'ORDONNEE (R) DES POINTS
*             DE GAUSS (QUADILATERE QUELCONQUE)
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER IDIM,ITEMP,NCOORD
      COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
      INTEGER NPGAUS
      REAL*8 XGAUSS(NPGAUS,NCOORD),R(NPGAUS)
      REAL*8 COORDE(9,2),Z
      REAL*8 L,L1,L2,H,H1,H2,H3
      INTEGER G
*
* VERIFICATION DU TYPE ET DU NOMBRE DE COORDONNES
*
      IF(IDIM.EQ.0.OR.IDIM.EQ.3.OR.NCOORD.EQ.1) THEN
        DO 10 G=1,NPGAUS
          R(G)=1.0D0
 10     CONTINUE
        RETURN
      ELSE
*
* CALCUL DES RAPPORTS GEOMETRIQUES DE L'ELEMENT
*
      L1=COORDE(4,2)-COORDE(1,2)
      L2=COORDE(3,2)-COORDE(2,2)
      H1=COORDE(4,1)-COORDE(1,1)
      H2=COORDE(3,1)-COORDE(4,1)
      H3=COORDE(2,1)-COORDE(3,1)
*
* CALCUL DE L'ABSCISSE DES POINTS DE GAUSS
*
```

```
      DO 20 G=1,NPGAUS
        H= (H1+H2+H3)-XGAUSS(G,2)*((H1/L1)+(H3/L2))
        Z=COORDE(1,1)+XGAUSS(G,2)*(H1/L1)+(H/2.0D0)*(1.0D0+XGAUSS(G,1))
*
* CALCUL DE L'ORDONNEE DES POINTS DE GAUSS
*
        IF(Z.GE.COORDE(1,1).AND.Z.LT.COORDE(4,1).AND.H1.GT.0.0D0) THEN
          L=(L1/H1)*(H1/2.0D0)*(1+XGAUSS(G,1))
        ELSEIF(Z.GT.COORDE(3,1).AND.Z.LE.COORDE(2,1).AND.H3.GT.0.0D0)
     &  THEN
          L=(L2/H3)*(H3/2.0D0)*(1-XGAUSS(G,1))
        ELSE
          L=L1+((L2-L1)/H2)*(H2/2.0D0)*(1+XGAUSS(G,1))
        ENDIF
        R(G)=COORDE(1,2)+(L/2.0D0)*(1.0D0+XGAUSS(G,2))
 20   CONTINUE
      ENDIF
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE RHEO(ETA,DRGLB,SHAPE,LM,NPGAUS,ITER,
     &                NBREQ,NBRCNT,NBRDLE,X,CONTNT,R)
*     ================================================================
*
*  FONCTION : CALCUL DE LA VISCOSITE
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      COMMON /BLOC/ TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      INTEGER IDIM,ITEMP,NCOORD
      COMMON/ CONTRL1/ IDIM,ITEMP,NCOORD
      REAL*8 RHO,VISCOS,CP,COND
      COMMON /PROP/ RHO,VISCOS,CP,COND
      INTEGER NPGAUS,ITER
      INTEGER NBREQ,NBRCNT,NBRDLE
      INTEGER LM(NBRDLE),I,J,G,POSI
      REAL*8 ETA(NPGAUS),DRGLB(NPGAUS,NDP,NCOORD),SHAPE(NPGAUS,NDP)
      REAL*8 X(NBREQ),CONTNT(NBRCNT),RESULT(9,2)
      REAL*8 R(NPGAUS)
      REAL*8 GAMMA,DUDY,DVDX,DUDX,DVDY,V
      REAL*8 T,S,M0,N
      REAL*8 AXI
*
*  PARAMETRES DES MODELES
*
      M0=1.875D5
      T=1.1345D0
      N=0.5D0
      S=(1.0D0-N)/2.0D0
      DO 50 G=1,NPGAUS
 50   ETA(G)=VISCOS
*
*  TEST COORDONNEES CARTESIENNES OU CYLINDRIQUES
*
      IF(IDIM.EQ.1.OR.IDIM.EQ.2) THEN
        AXI=1.0D0
      ELSEIF(IDIM.EQ.0.OR.IDIM.EQ.3) THEN
        AXI=0.0D0
      ENDIF
*
      IF(MVISC.NE.0.AND.ITER.GT.1) THEN
*
*  RECUPERATION DU PROFIL DE VITESSE DE L'ELEMENT
*
```

```
        POSI=0
        DO 20 J=1,2
          DO 10 I=1,NDP
            POSI=POSI+1
            IF(LM(POSI).LT.0) THEN
              RESULT(I,J)=CONTNT(-LM(POSI))
            ELSEIF(LM(POSI).EQ.0) THEN
              RESULT(I,J)=0.0D0
            ELSEIF(LM(POSI).GT.0) THEN
              RESULT(I,J)=X(LM(POSI))
            ENDIF
 10       CONTINUE
 20     CONTINUE
*
*  CALCUL DE LA VITESSE DE CISAILLEMENT
*
        DO 100 G=1,NPGAUS
          DUDY=0.0D0
          DVDX=0.0D0
          DUDX=0.0D0
          DVDY=0.0D0
          V   =0.0D0
          DO 200 I=1,NDP
            DUDY=DUDY+RESULT(I,1)*DRGLB(G,I,2)
            DVDX=DVDX+RESULT(I,2)*DRGLB(G,I,1)
            DUDX=DUDX+RESULT(I,1)*DRGLB(G,I,1)
            DVDY=DVDY+RESULT(I,2)*DRGLB(G,I,2)
            V   =V   +RESULT(I,2)*SHAPE(G,I)
 200       CONTINUE
           GAMMA=DSQRT(2.0D0*DUDX**2+2.0D0*DVDY**2+(DUDY+DVDX)**2
     &                 +AXI*2.0D0*(V/R(G))**2)
*
*  LOI DE PUISSANCE
*
            IF(MVISC.EQ.1) THEN
              IF(GAMMA.GT.0.001) THEN
                ETA(G)=M0*GAMMA**(N-1.0D0)
              ELSE
                ETA(G)=M0*(1.0D-03)**(N-1.0D0)
              ENDIF
            ENDIF
*
*  MODELE DE CARREAU A TROIS PARAMETRES
*
            IF(MVISC.EQ.2) THEN
              ETA(G)=ETA(G)/((1.0D0+(T*GAMMA)**2)**S)
            ENDIF
 100     CONTINUE
       ENDIF
       RETURN
       END
*
*
*      ================================================================
       SUBROUTINE MATDIF(SHAPE,DSGLB,DJACV,ETA,POIDS,
     &                   MATELM,NPGAUS,NDP,MELDOF,R)
*      ================================================================
*
*  FONCTION : CALCULE LES CONTRIBUTIONS A LA MATRICE ELEMENTAIRE DES TERMES
*             DIFFUSSIFS
*
       IMPLICIT CHARACTER (A-Z)
       INTEGER IPRINT
       COMMON / DISKS / IPRINT
       INTEGER IDIM,ITEMP,NCOORD
       COMMON / CONTRL1 / IDIM,ITEMP,NCOORD
```

```
      REAL*8 DE
      COMMON / DEBORAH / DE
      INTEGER NPGAUS,NDP,MELDOF
      INTEGER I,J,K
      REAL*8 ETA(NPGAUS),MATELM(MELDOF,MELDOF)
      REAL*8 DSGLB(NPGAUS,NDP,NCOORD),DJACV(NPGAUS)
      REAL*8 SHAPE(NPGAUS,NDP),POIDS(NPGAUS)
      REAL*8 R(NPGAUS)
      REAL*8 K11,K12,K22
      REAL*8 AXI
*
*  TEST CORDONNEES CARTESIENNES OU CYLINDRIQUES
*
      IF(IDIM.EQ.1.OR.IDIM.EQ.2) THEN
        AXI=1.0D0
      ELSEIF(IDIM.EQ.0.OR.IDIM.EQ.3) THEN
        AXI=0.0D0
      ENDIF
*
*  CALCUL DE LA MATRICE ELEMENTAIRE TERME DIFFUSIF
*  EQUATION DE STOKES
*
      DO 930 I=1,NDP
        DO 931 J=1,NDP
           K11=0.0D0
           K12=0.0D0
           K22=0.0D0
           DO 932 K=1,NPGAUS
             K11=K11+POIDS(K)*DJACV(K)*ETA(K)*(2.0D0*R(K)*DSGLB(K,I,1)*
     &               DSGLB(K,J,1)+R(K)*DSGLB(K,I,2)*DSGLB(K,J,2))
             K12=K12+POIDS(K)*ETA(K)*DSGLB(K,I,2)*DSGLB(K,J,1)*R(K)*
     &              DJACV(K)
             K22=K22+POIDS(K)*ETA(K)*(R(K)*DSGLB(K,I,1)*DSGLB(K,J,1)+
     &              2.0D0*R(K)*DSGLB(K,I,2)*DSGLB(K,J,2)+AXI*2.0D0*
     &               SHAPE(K,I)*SHAPE(K,J)/R(K))*DJACV(K)
 932       CONTINUE
           MATELM(I,J)         =DE*K11
           MATELM(I,J+NDP)     =DE*K12
           MATELM(J+NDP,I)     =DE*K12
           MATELM(I+NDP,J+NDP)=DE*K22
 931     CONTINUE
 930   CONTINUE
       RETURN
       END
*
*     ================================================================
      SUBROUTINE MATPEN(SHAPE,DSGLB,DJACP,E,POIDS,
     &                  MATELM,NPGAUS,NDP,MELDOF,R)
*     ================================================================
*
*  FONCTION : CALCULE LES CONTRIBUTIONS A LA MATRICE ELEMENTAIRE DES TERMES
*             DE PENALISATION
*
      IMPLICIT NONE
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      INTEGER IDIM,ITEMP,NCOORD
      COMMON / CONTRL1/ IDIM,ITEMP,NCOORD
      REAL*8 DE
      COMMON / DEBORAH / DE
      INTEGER I,J,K,NPGAUS,NDP,MELDOF
      REAL*8 E,MATELM(MELDOF,MELDOF),POIDS(NPGAUS)
      REAL*8 DSGLB(NPGAUS,NDP,NCOORD),DJACP(NPGAUS)
      REAL*8 SHAPE(NPGAUS,NDP)
      REAL*8 R(NPGAUS)
      REAL*8 K11,K12,K22
```

```
      REAL*8 AXI
*
*  TEST COORDONNEES CARTESIENNES OU CYLINDRIQUES
*
      IF(IDIM.EQ.1.OR.IDIM.EQ.2) THEN
        AXI=1.0D0
      ELSEIF(IDIM.EQ.0.OR.IDIM.EQ.3) THEN
        AXI=0.0D0
      ENDIF
*
*  CALCUL DE LA MATRICE ELEMENTAIRE TERME DIFFUSIF
*  EQUATION DE STOKE
*
      DO 940 I=1,NDP
        DO 941 J=1,NDP
          K11=0.0D0
          K12=0.0D0
          K22=0.0D0
          DO 942 K=1,NPGAUS
            K11=K11+POIDS(K)*DJACP(K)*(R(K)*DSGLB(K,I,1)*DSGLB(K,J,1))
            K12=K12+POIDS(K)*DJACP(K)*(R(K)*DSGLB(K,I,1)*DSGLB(K,J,2)
     &             +AXI*SHAPE(K,J)*DSGLB(K,I,1))
            K22=K22+POIDS(K)*DJACP(K)*(R(K)*DSGLB(K,I,2)*DSGLB(K,J,2)+
     &              AXI*(SHAPE(K,J)*DSGLB(K,I,2)+SHAPE(K,I)*
     &              DSGLB(K,J,2)+SHAPE(K,J)*SHAPE(K,I)/R(K)))
 942      CONTINUE
          MATELM(I,J)=MATELM(I,J)+K11*(DE/E)
          MATELM(I,J+NDP)=MATELM(I,J+NDP)+K12*(DE/E)
          MATELM(J+NDP,I)=MATELM(J+NDP,I)+K12*(DE/E)
          MATELM(I+NDP,J+NDP)=MATELM(I+NDP,J+NDP)+K22*(DE/E)
 941    CONTINUE
 940  CONTINUE
      RETURN
      END

*
*
*     ================================================================
      SUBROUTINE SIGMA(ITERGL,SXX,SYY,SXY,NPGAUS,NPGAUS1,NDP,NDP1,
     &                 LM,LM1,X,X1,CONTNT,CONTNT1,SHAPE1,DSGLB,POIDS1)
*     ================================================================
*
*     FONCTION: CALCUL DU TENSEUR DES CONTRAINTES VISCOELASTIQUES
*               DE L'ELEMENT
*
*     VARIABLES GLOBALES
*     ------------------
      IMPLICIT NONE
      INTEGER IDIM,ITEMP,NCOORD
      COMMON/ CONTRL1 / IDIM,ITEMP,NCOORD
      INTEGER NFIXB,IFIX,NFIXB1,IFIX1
      COMMON/ FIX / NFIXB, IFIX(4), NFIXB1, IFIX1(3)
      INTEGER NEQT, MELDOF, ESPMAT
      COMMON/ MATDIM / NEQT, MELDOF, ESPMAT
      INTEGER NEQT1, MELDOF1, ESPMAT1
      COMMON/ MATDIM1 / NEQT1, MELDOF1, ESPMAT1
      REAL*8 RHO,VISCOS,CP,COND
      COMMON / PROP / RHO,VISCOS,CP,COND
      REAL*8 DE
      COMMON / DEBORAH / DE
      INTEGER NEWTON
      COMMON / PREMIERE / NEWTON
*
*     ARGUMENTS
*     ---------
      INTEGER ITERGL,NPGAUS,NPGAUS1,NDP,NDP1
```

```
      INTEGER LM(MELDOF),LM1(MELDOF1)
      REAL*8 POIDS1(NPGAUS1),SHAPE1(NPGAUS1,NDP1)
      REAL*8 DSGLB(NPGAUS,NDP,NCOORD)
      REAL*8 X(NEQT),X1(NEQT1),CONTNT(NFIXB),CONTNT1(NFIXB1)
      REAL*8 SXX(NPGAUS),SYY(NPGAUS),SXY(NPGAUS)
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER I,J,K,G,POSI
      REAL*8 DUDY,DVDX,DUDX,DVDY,CXX,CYY,CXY
      REAL*8 RESULT(9,2),RESULT1(4,3),PARENT,f
      REAL*8 XGAUSS(9,2),WGAUSS(9),SHAPE94(9,4),DSLOC94(9,4,2)
*
      IF(NEWTON.EQ.0) THEN
*
*     PREMIERE ITERATION NEWTONNIENNE (SIJ=0)
*
        IF(ITERGL.EQ.1) THEN
          DO 55 G=1,NPGAUS
            SXX(G)=0.0D0
            SYY(G)=0.0D0
            SXY(G)=0.0D0
 55       CONTINUE
          RETURN
        ENDIF
      ENDIF
*
*     CALCUL DES FONCTIONS DE BASE NECESSAIRE A L'EVALUATION DU
*     TENSEUR DE CONFORMATION AUX POINTS DE GAUSS DE VITESSES
*
      CALL GAUSS(3,NCOORD,NPGAUS,XGAUSS,WGAUSS)
      CALL FBASE(NPGAUS,NDP1,NCOORD,XGAUSS,SHAPE94,DSLOC94,1)
*
*     RECUPERATION DU PROFIL DE VITESSES DE L'ELEMENT
*
      POSI=0
      DO 20 J=1,2
        DO 10 I=1,NDP
          POSI=POSI+1
          IF(LM(POSI).LT.0) THEN
            RESULT(I,J)=CONTNT(-LM(POSI))
          ELSEIF(LM(POSI).EQ.0) THEN
            RESULT(I,J)=0.0D0
          ELSEIF(LM(POSI).GT.0) THEN
            RESULT(I,J)=X(LM(POSI))
          ENDIF
 10     CONTINUE
 20   CONTINUE
*
*     RECUPERATION DU PROFIL DE TENSEUR DE CONFORMATION DE L'ELEMENT
*
      POSI=0
      DO 21 J=1,3
        DO 11 I=1,NDP1
          POSI=POSI+1
          IF(LM1(POSI).LT.0) THEN
            RESULT1(I,J)=CONTNT1(-LM1(POSI))
          ELSEIF(LM1(POSI).EQ.0) THEN
            RESULT1(I,J)=0.0D0
          ELSEIF(LM1(POSI).GT.0) THEN
            RESULT1(I,J)=X1(LM1(POSI))
          ENDIF
 11     CONTINUE
 21   CONTINUE
*
*     CALCUL DU GRADIENT DE VITESSE DE L'ELEMENT
```

```
*
        DO 100 G=1,NPGAUS
          DUDY=0.0D0
          DVDX=0.0D0
          DUDX=0.0D0
          DVDY=0.0D0
          DO 200 K=1,NDP
            DUDY=  DUDY+RESULT(K,1)*DSGLB(G,K,2)
            DVDX=  DVDX+RESULT(K,2)*DSGLB(G,K,1)
            DUDX=  DUDX+RESULT(K,1)*DSGLB(G,K,1)
            DVDY=  DVDY+RESULT(K,2)*DSGLB(G,K,2)
 200     CONTINUE
*
*       CALCUL DE LA VALEUR DU TENSEUR DE CONFORMATION AU 9 POINTS
*       DE GAUSS DE VITESSES
*
          CXX=0.0D0
          CYY=0.0D0
          CXY=0.0D0
          DO 201 K=1,NDP1
            CXX= CXX + RESULT1(K,1)*SHAPE94(G,K)
            CYY= CYY + RESULT1(K,2)*SHAPE94(G,K)
            CXY= CXY + RESULT1(K,3)*SHAPE94(G,K)
 201      CONTINUE
          PARENT=(DUDY*CXY+DVDX*CXY+DUDX*CXX+DVDY*CYY)
*
*      CALCUL DES COMPOSANTES VISCOELASTIQUES DU TENSEUR DES CONTRAINTES
*
          f=1.0d0
          SXX(G)=1.0D0-3.0D0*CXX-2.0D0*DE*PARENT*CXX+2.0D0*f*DE*VISCOS*DUDX
          SYY(G)=1.0D0-3.0D0*CYY-2.0D0*DE*PARENT*CYY+2.0D0*f*DE*VISCOS*DVDY
          SXY(G)=      -3.0D0*CXY-2.0D0*DE*PARENT*CXY+f*DE*VISCOS*(DUDY+DVDX)
 100   CONTINUE
       RETURN
       END
*
*
*      ================================================================
       SUBROUTINE MEMBD(BELM,NBRDLE,SXX,SYY,SXY,DSGLB,
      &                 NPGAUS,NDP,DJAC,POIDS)
*      ================================================================
*
*      FONCTION: CONTRIBUTION DES CONTRAINTES VISCOELASTIQUES AU
*                MEMBRE DE DROITE DU VECTEUR B ELEMENTAIRE
*
       IMPLICIT CHARACTER (A-Z)
       INTEGER IDIM,ITEMP,NCOORD
       COMMON / CONTRL1 / IDIM,ITEMP,NCOORD
       INTEGER NBRDLE,NPGAUS,NDP
       REAL*8 BELM(NBRDLE),SXX(NPGAUS),SYY(NPGAUS),SXY(NPGAUS)
       REAL*8 DSGLB(NPGAUS,NDP,NCOORD),DJAC(NPGAUS)
       REAL*8 POIDS(NPGAUS)
       INTEGER I,G
       REAL*8 B1,B2
*
       DO 20 I=1,NDP
         B1=0.0D0
         B2=0.0D0
         DO 10 G=1,NPGAUS
           B1=B1+(DSGLB(G,I,2)*SXY(G)+DSGLB(G,I,1)*SXX(G))*
      &          POIDS(G)*DJAC(G)
           B2=B2+(DSGLB(G,I,1)*SXY(G)+DSGLB(G,I,2)*SYY(G))*
      &          POIDS(G)*DJAC(G)
 10      CONTINUE
         BELM(I)       =  -B1
         BELM(I+NDP)   =  -B2
```

```
 20   CONTINUE
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE CONVERGE(ITERGL,NELEM,IELEMG,NPGAUS,SXX,SYY,SXY,
     &                    SXXIM1,SYYIM1,SXYIM1,ECART,NORMS,CONVRG,
     &                    COMPTEUR)
*     ================================================================
*
*  FONCTION: CALCUL DE L'ECART DU CHAMP DES CONTRAINTES VISCOELASTIQUES
*            POUR DEUX ITERATIONS SUCCESSIVES
*
*
*
*     ARGUMENTS
*     ---------
      INTEGER ITERGL, NELEM, IELEMG, NPGAUS, COMPTEUR
      REAL*8 SXXIM1(NELEM,NPGAUS), SYYIM1(NELEM,NPGAUS),
     &       SXYIM1(NELEM,NPGAUS)
      REAL*8 SXX(NPGAUS), SYY(NPGAUS), SXY(NPGAUS)
      REAL*8 ECART,NORMS
      LOGICAL CONVRG
*
*     VARIABLE LOCALE
*     ---------------
      INTEGER G
      REAL*8 DELTA
*
      IF(ITERGL.EQ.1) THEN
        DO 60 G = 1, NPGAUS
          SXXIM1(IELEMG,G) = 0.0D0
          SYYIM1(IELEMG,G) = 0.0D0
          SXYIM1(IELEMG,G) = 0.0D0
 60     CONTINUE
      ENDIF
      DO 70 G = 1, NPGAUS
        DELTA =          DABS(SXX(G) - SXXIM1(IELEMG,G))
     &                 + DABS(SYY(G) - SYYIM1(IELEMG,G))
     &                 + DABS(SXY(G) - SXYIM1(IELEMG,G))
        IF(DELTA.GT.1.0D-04) THEN
          CONVRG = .FALSE.
          COMPTEUR =COMPTEUR + 1
        ENDIF
        ECART = ECART + DELTA
        NORMS = NORMS + DABS(SXX(G)) + DABS(SYY(G)) + DABS(SXY(G))
        SXXIM1(IELEMG,G) = SXX(G)
        SYYIM1(IELEMG,G) = SYY(G)
        SXYIM1(IELEMG,G) = SXY(G)
 70   CONTINUE
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE VISCO(SHAPE,SHAPE1,DJAC,DSGLB,DSGLB1,POIDS,POIDS1,
     &                 MATELM1,NPGAUS,NPGAUS1,NDP,NDP1,LM,LM1,X,X1,
     &                 CONTNT,CONTNT1,COORDE)
*     ================================================================
*
*     FONCTION: CALCUL DE LA CONTRIBUTION DES TERMES ASSOCIES AU
*               TENSEUR DE CONFORMATION A LA MATRICE ELEMENTAIRE
*
*
*     VARIABLES GLOBALES
```

```
*     ------------------
      IMPLICIT NONE
      INTEGER IDIM,ITEMP,NCOORD
      COMMON/ CONTRL1 / IDIM,ITEMP,NCOORD
      INTEGER NFIXB,IFIX,NFIXB1,IFIX1
      COMMON/ FIX / NFIXB,IFIX(4),NFIXB1,IFIX1(3)
      INTEGER NEQT,MELDOF,ESPMAT
      COMMON/ MATDIM / NEQT,MELDOF,ESPMAT
      INTEGER NEQT1,MELDOF1,ESPMAT1
      COMMON/ MATDIM1 / NEQT1,MELDOF1,ESPMAT1
      REAL*8 DE
      COMMON / DEBORAH / DE
      INTEGER UPWIND
      COMMON / UPWINDING / UPWIND
*
*     ARGUMENTS
*     ---------
      INTEGER NPGAUS,NPGAUS1,NDP,NDP1
      INTEGER LM(MELDOF),LM1(MELDOF1)
      REAL*8 SHAPE(NPGAUS,NDP),SHAPE1(NPGAUS1,NDP1)
      REAL*8 MATELM1(MELDOF1,MELDOF1),X(NEQT),X1(NEQT1)
      REAL*8 CONTNT(NFIXB),CONTNT1(NFIXB1)
      REAL*8 POIDS(NPGAUS),POIDS1(NPGAUS1),DJAC(NPGAUS1)
      REAL*8 DSGLB(NPGAUS,NDP,NCOORD),DSGLB1(NPGAUS1,NDP1,NCOORD)
      REAL*8 COORDE(9,2)
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER I,J,K,G,POSI
      REAL*8 RESULT(9,2),RESULT1(4,3),PARENT
      REAL*8 U,V,DUDY,DVDX,DUDX,DVDY,CXX,CYY,CXY
      REAL*8 K33,K34,K35,K43,K44,K45,K53,K54,K55
      REAL*8 XGAUSS(4,2),WGAUSS(4),SHAPE49(4,9),DSLOC49(4,9,2)
      REAL*8 DSGLB49(4,9,2),DJAC4(4)
      REAL*8 SHAPESU(4,4)
      REAL*8 H1,H2,L1,L2,HKSI,HETA,VKSI,VETA,KBAR,som
*
*     CALCUL DES FONCTIONS DE BASE ET DES DERIVEES NECESSAIRES
*     A L'EVALUATION DES VITESSES ET GRADIENTS DE VITESSES
*     AUX POINTS DE GAUSS DE CONFORMATION
*
      CALL GAUSS(2,NCOORD,NPGAUS1,XGAUSS,WGAUSS)
      CALL FBASE(NPGAUS1,NDP,NCOORD,XGAUSS,SHAPE49,DSLOC49,2)

*
*     RECUPERATION DU PROFIL DE VITESSE DE L'ELEMENT
*
      POSI=0
      DO 20 J=1,2
          DO 10 I=1,NDP
            POSI=POSI+1
            IF(LM(POSI).LT.0) THEN
              RESULT(I,J)=CONTNT(-LM(POSI))
            ELSEIF(LM(POSI).EQ.0) THEN
              RESULT(I,J)=0.0D0
            ELSEIF(LM(POSI).GT.0) THEN
              RESULT(I,J)=X(LM(POSI))
            ENDIF
 10       CONTINUE
 20     CONTINUE
      CALL DRGLB(NPGAUS1,NDP,NCOORD,COORDE,DSLOC49,DSGLB49,DJAC4)
*
*     CALCUL DU POIDS POUR L'UPWINDING (CONSTANT SUR L'ELEMENT)
*
      L1=COORDE(2,1)-COORDE(1,1)
      L2=COORDE(3,1)-COORDE(4,1)
```

```
      H1=COORDE(4,2)-COORDE(1,2)
      H2=COORDE(3,2)-COORDE(2,2)
      HKSI=(L1+L2)/2.0D0
      HETA=(H1+H2)/2.0D0
      VKSI=RESULT(9,1)
      VETA=RESULT(9,2)
      KBAR=(DABS(VKSI*HKSI)+DABS(VETA*HETA))/2.0D0
*
*     RECUPERATION DU PROFIL DE TENSEUR DE CONFORMATION DE L'ELEMENT
*
      POSI=0
      DO 21 J=1,3
          DO 11 I=1,NDP1
            POSI=POSI+1
            IF(LM1(POSI).LT.0) THEN
              RESULT1(I,J)=CONTNT1(-LM1(POSI))
            ELSEIF(LM1(POSI).EQ.0) THEN
              RESULT1(I,J)=0.0D0
            ELSEIF(LM1(POSI).GT.0) THEN
              RESULT1(I,J)=X1(LM1(POSI))
            ENDIF
 11       CONTINUE
 21   CONTINUE
*
*     CALCUL DES COMPOSANTES DU TENSEUR DE CONFORMATION
*
      DO 50 I=1,NDP1
        DO 60 J=1,NDP1
          K33=0.0D0
          K34=0.0D0
          K35=0.0D0
          K43=0.0D0
          K44=0.0D0
          K45=0.0D0
          K53=0.0D0
          K54=0.0D0
          K55=0.0D0
*
          som=0.0d0
*
          DO 101 G=1,NPGAUS1
            CXX=0.0D0
            CYY=0.0D0
            CXY=0.0D0
            DO 201 K=1,NDP1
              CXX=CXX+RESULT1(K,1)*SHAPE1(G,K)
              CYY=CYY+RESULT1(K,2)*SHAPE1(G,K)
              CXY=CXY+RESULT1(K,3)*SHAPE1(G,K)
 201        CONTINUE
*
*     CALCUL DES VITESSES ET GRADIENTS DE VITESSES AUX 4 POINTS
*     DE GAUSS DE CONFORMATION
*
         DUDY=0.0D0
         DVDX=0.0D0
         DUDX=0.0D0
         DVDY=0.0D0
            U=0.0D0
            V=0.0D0
            DO 200 K=1,NDP
              DUDY=  DUDY+RESULT(K,1)*DSGLB49(G,K,2)
              DVDX=  DVDX+RESULT(K,2)*DSGLB49(G,K,1)
              DUDX=  DUDX+RESULT(K,1)*DSGLB49(G,K,1)
              DVDY=  DVDY+RESULT(K,2)*DSGLB49(G,K,2)
                 U=     U+RESULT(K,1)*SHAPE49(G,K)
                 V=     V+RESULT(K,2)*SHAPE49(G,K)
```

```
 200        CONTINUE
*
*      CALCUL DU POIDS POUR L'UPWINDING
*      UPWIND=0 => GALERKIN
*      UPWIND=1 => SU
*      UPWIND=2 => SUPG
*
           IF(UPWIND.EQ.0.OR.UPWIND.EQ.1) THEN
           SHAPESU(G,J)=SHAPE1(G,J)+DFLOAT(UPWIND)*(KBAR/(U*U+V*V))*
     &                  (U*DSGLB1(G,J,1)+V*DSGLB1(G,J,2))
           ELSE
           SHAPESU(G,J)=SHAPE1(G,J)+(KBAR/(U*U+V*V))*
     &                  (U*DSGLB1(G,J,1)+V*DSGLB1(G,J,2))
           ENDIF
*
*      CALCUL DES CONTRIBUTIONS A LA MATRICE ELEMENTAIRE
*
           PARENT=DUDY*CXY+DVDX*CXY+DUDX*CXX+DVDY*CYY
*
           IF(UPWIND.EQ.0.OR.UPWIND.EQ.1) THEN
*
           K33=K33+POIDS1(G)*DJAC(G)*(SHAPESU(G,J)*DE*(dudx*shape1(G,I)
     &        +dvdy*shape1(G,I))+SHAPE1(G,J)*DE*(2.0D0*DUDX*SHAPE1(G,I)
     &        -2.0D0*PARENT*SHAPE1(G,I))-3.0D0*SHAPE1(G,J)*SHAPE1(G,I))
           K34=0.0D0
           K35=K35+POIDS1(G)*DJAC(G)*DE*SHAPE1(G,J)*(2.0D0*DUDY*
     &           SHAPE1(G,I))
           K43=0.0D0
           K44=K44+POIDS1(G)*DJAC(G)*(SHAPESU(G,J)*DE*(dudx*shape1(G,I)
     &        +dvdy*shape1(G,I))+SHAPE1(G,J)*DE*(2.0D0*DVDY*SHAPE1(G,I)
     &        -2.0D0*PARENT*SHAPE1(G,I))-3.0D0*SHAPE1(G,J)*SHAPE1(G,I))
           K45=K45+POIDS1(G)*DJAC(G)*DE*SHAPE1(G,J)*(2.0D0*DVDX*
     &           SHAPE1(G,I))
           K53=K53+POIDS1(G)*DJAC(G)*DE*SHAPE1(G,J)*(DVDX*SHAPE1(G,I))
           K54=K54+POIDS1(G)*DJAC(G)*DE*SHAPE1(G,J)*(DUDY*SHAPE1(G,I))
           K55=K55+POIDS1(G)*DJAC(G)*(SHAPESU(G,J)*DE*(dudx*shape(G,I)
     &         +dvdy*shape1(G,I))+SHAPE1(G,J)*DE*(DUDX+DVDY
     &        -2.0D0*PARENT)*SHAPE1(G,I)-3.0D0*SHAPE1(G,J)*SHAPE1(G,I))
*
           som=som+POIDS1(G)*DJAC(G)*SHAPESU(G,J)*DE*(dudx*shape1(g,i)
     &                                              +dvdy*shape1(g,i))
*
           ELSE
*
           K33=K33+POIDS1(G)*DJAC(G)*(SHAPESU(G,J)*DE*(dudx*shape1(G,I)
     &       +dvdy*shape1(G,I))+SHAPESU(G,J)*DE*(2.0D0*DUDX*SHAPE1(G,I)
     &       -2.0D0*PARENT*SHAPE1(G,I))-3.0D0*SHAPESU(G,J)*SHAPE1(G,I))
           K34=0.0D0
           K35=K35+POIDS1(G)*DJAC(G)*DE*SHAPESU(G,J)*(2.0D0*DUDY*
     &           SHAPE1(G,I))
           K43=0.0D0
           K44=K44+POIDS1(G)*DJAC(G)*(SHAPESU(G,J)*DE*(dudx*shape1(G,I)
     &       +dvdy*shape1(G,I))+SHAPESU(G,J)*DE*(2.0D0*DVDY*SHAPE1(G,I)
     &       -2.0D0*PARENT*SHAPE1(G,I))-3.0D0*SHAPESU(G,J)*SHAPE1(G,I))
           K45=K45+POIDS1(G)*DJAC(G)*DE*SHAPESU(G,J)*(2.0D0*DVDX*
     &           SHAPE1(G,I))
           K53=K53+POIDS1(G)*DJAC(G)*DE*SHAPESU(G,J)*(DVDX*SHAPE1(G,I))
           K54=K54+POIDS1(G)*DJAC(G)*DE*SHAPESU(G,J)*(DUDY*SHAPE1(G,I))
           K55=K55+POIDS1(G)*DJAC(G)*(SHAPESU(G,J)*DE*(dudx*shape1(G,I)
     &       +dvdy*shape1(G,I))+SHAPESU(G,J)*DE*(DUDX+DVDY
     &       -2.0D0*PARENT)*SHAPE1(G,I)-3.0D0*SHAPESU(G,J)*SHAPE1(G,I))

           ENDIF
 101    CONTINUE
*
*          write(2,*) 'som=',som
```

```
*
          MATELM1(I       ,J       )=K33
          MATELM1(I       ,J+  NDP1)=K34
          MATELM1(I       ,J+2*NDP1)=K35
          MATELM1(I+  NDP1,J       )=K43
          MATELM1(I+  NDP1,J+  NDP1)=K44
          MATELM1(I+  NDP1,J+2*NDP1)=K45
          MATELM1(I+2*NDP1,J       )=K53
          MATELM1(I+2*NDP1,J+  NDP1)=K54
          MATELM1(I+2*NDP1,J+2*NDP1)=K55
 60     CONTINUE
 50   CONTINUE
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE MEMBD1(BELM1,NPGAUS1,SHAPE1,NDP,NDP1,DJAC,POIDS1,
     &                  LM,LM1,X,X1,CONTNT,CONTNT1,COORDE1,ITERGL)
*     ================================================================
*
*  FONCTION: CALCUL DE LA CONTRIBUTION AU MEMBRE DE DROITE POUR
*            LE SYSTEME ELEMENTAIRE A1*X1=B1 ET CALCUL DE L'INTEGRALE
*            DE BORD A LA SORTIE
*
*
*     VARIABLES GLOBALES
*     ------------------
      IMPLICIT NONE
      INTEGER NFIXB,IFIX,NFIXB1,IFIX1
      COMMON/ FIX / NFIXB,IFIX(4),NFIXB1,IFIX1(3)
      INTEGER NEQT, MELDOF, ESPMAT
      COMMON/ MATDIM /  NEQT, MELDOF, ESPMAT
      INTEGER NEQT1,MELDOF1,ESPMAT1
      COMMON/ MATDIM1 / NEQT1,MELDOF1,ESPMAT1
      REAL*8 DE
      COMMON/ DEBORAH / DE
      REAL*8 XMAX
      COMMON/ SORTIE / XMAX
*
*     ARGUMENTS
*     ---------
      INTEGER NPGAUS1,NDP1,NDP,ITERGL
      INTEGER LM(MELDOF),LM1(MELDOF1)
      REAL*8 SHAPE1(NPGAUS1,NDP1)
      REAL*8 BELM1(MELDOF1),DJAC(NPGAUS1),POIDS1(NPGAUS1)
      REAL*8 X(NEQT),  CONTNT(NFIXB)
      REAL*8 X1(NEQT1),CONTNT1(NFIXB1)
      REAL*8 COORDE1(4,2)
*
*     VARIABLES LOCALES
*     -----------------
      INTEGER I,J,K,G,POSI
      REAL*8 B1,B2,B3,RESULT1(4,3),RESULT(9,2),H,F2,F3,W
*
*     CALCUL DU MEMBRE DE DROITE
*     --------------------------
*
      DO 50 I=1,NDP1
        B1=0.0D0
        B2=0.0D0
        B3=0.0D0
        DO 101 G=1,NPGAUS1
          B1=B1+(1.0D0)*SHAPE1(G,I)*POIDS1(G)*DJAC(G)
          B2=B2+(1.0D0)*SHAPE1(G,I)*POIDS1(G)*DJAC(G)
          B3=0.0D0
```

```
 101    CONTINUE
        BELM1(I       ) = - B1
        BELM1(I+  NDP1) = - B2
        BELM1(I+2*NDP1) = - B3
 50   CONTINUE
*
*     CALCUL DE L'INTEGRALE DE BORD AU PLAN DE SORTIE
*     -----------------------------------------------
*
      IF(DABS(COORDE1(3,1)-XMAX).LT.1.0D-10.AND.ITERGL.GE.1) THEN
*
*     RECUPERATION DU PROFIL DE VITESSE DE L'ELEMENT
*
      POSI=0
      DO 20 J=1,2
          DO 10 I=1,NDP
            POSI=POSI+1
            IF(LM(POSI).LT.0) THEN
              RESULT(I,J)=CONTNT(-LM(POSI))
            ELSEIF(LM(POSI).EQ.0) THEN
              RESULT(I,J)=0.0D0
            ELSEIF(LM(POSI).GT.0) THEN
              RESULT(I,J)=X(LM(POSI))
            ENDIF
 10       CONTINUE
 20   CONTINUE
*
*     RECUPERATION DU PROFIL DE TENSEUR DE CONFORMATION DE L'ELEMENT
*
      POSI=0
      DO 21 J=1,3
          DO 11 I=1,NDP1
            POSI=POSI+1
            IF(LM1(POSI).LT.0) THEN
              RESULT1(I,J)=CONTNT1(-LM1(POSI))
            ELSEIF(LM1(POSI).EQ.0) THEN
              RESULT1(I,J)=0.0D0
            ELSEIF(LM1(POSI).GT.0) THEN
              RESULT1(I,J)=X1(LM1(POSI))
            ENDIF
 11       CONTINUE
 21   CONTINUE
*
*     CALCUL DES TERMES DE "TRACTION" PAR LA METHODE DU TRAPEZE
*
      H=COORDE1(3,2)-COORDE1(2,2)
      W=0.0D0
      DO 31 G=1,NPGAUS1
        W=W+DJAC(G)/4.0D0
 31   CONTINUE
      DO 30 J=1,3
        F2=W*DE*RESULT(2,1)*RESULT1(2,J)*H/2.0D0
        F3=W*DE*RESULT(3,1)*RESULT1(3,J)*H/2.0D0
        BELM1(2+(J-1)*NDP1) = BELM1(2+(J-1)*NDP1) + F2
        BELM1(3+(J-1)*NDP1) = BELM1(3+(J-1)*NDP1) + F3
 30   CONTINUE
      ENDIF
      RETURN
      END
*
*
*
*     ================================================================
      SUBROUTINE MISAJR(X,DX,NBREQ,FLAG)
*     ================================================================
*
```

```
*  FONCTION : MISE A JOUR DU VECTEUR SOLUTION
*
*  ARGUMENTS :
*    X ......... VECTEUR SOLUTION
*    DX ........ CORRECTION DU VECTEUR SOLUTION
*    FLAG ...... 0 => FACTEUR DE SOUS-CORRECTION
*         ...... 1 => PAS DE SOUS-CORRECTION
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      INTEGER NBREQ
      INTEGER I, FLAG
      REAL*8 X(NBREQ),DX(NBREQ), FACTEUR
*
*     ON UTILISE UN FACTEUR DE SOUS-CORRECTION
*     AFIN AUGMENTER LA VITESSE DE CONVERGENCE LORSQUE
*     LES CONTRAINTES VISCOELASTIQUES SONT NON NULLES
*
      IF(FLAG.EQ.0) THEN
        FACTEUR=1.0D0/2.0D0
      ELSEIF(FLAG.EQ.1) THEN
        FACTEUR=1.0D0
      ELSE
        WRITE(IPRINT,1001) FLAG
        STOP
      ENDIF
      DO 10 I=1,NBREQ
        X(I)=X(I)+FACTEUR*DX(I)
 10   CONTINUE
 1001 FORMAT(/,' ERREUR DANS "MISAJR", FLAG =',I3,' DOIT ETRE 0 OU 1')
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE OUTPUT
*     ================================================================
*
*  FONCTION : DRIVER D'IMPRESSION DES RESULTATS
*
      IMPLICIT NONE
      INTEGER MBLANK,MAXVAR
      PARAMETER (MBLANK=2500000,MAXVAR=4)
      INTEGER IA
      COMMON /BLANK/ IA(MBLANK)
      INTEGER IDIM,ITEMP,NCOORD
      COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
      INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
      COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                 NUMNP1,NNPE1,NDEP1
      INTEGER BLOC
      REAL*8 PROP
      COMMON /PARAM/ PROP(5,4),BLOC(5,8)
      INTEGER TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      COMMON /BLOC/ TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      REAL*8 RHO,VISCOS,CP,COND
      COMMON /PROP/ RHO,VISCOS,CP,COND
      INTEGER NEQT,MELDOF,ESPMAT
      COMMON /MATDIM/ NEQT,MELDOF,ESPMAT
      INTEGER NEQT1,MELDOF1,ESPMAT1
      COMMON /MATDIM1/ NEQT1, MELDOF1,ESPMAT1
*
      INTEGER NX,NNCOORD,NIDEQ,NCONSTR,NP,NC,NINTGP, NINTGU
      INTEGER NX1,NNCOORD1,NIDEQ1,NCONSTR1,NLMGLOB1,NCONNEC1
      INTEGER NN1
```

```
      INTEGER NLMGLOB,NCONNEC,IGR,IELEMG,IELEMG1
      INTEGER NXIM1,NX1IM1
      INTEGER ADDTBL,LOCTBL,SIZEOF,STOROF
      INTEGER TYPE1, NDP1, INTGC
      TYPE1=1
      NDP1= 4
      INTGC=2

*
      NINTGU=INTGV**NCOORD
      NINTGP=INTGP**NCOORD
      NN1=ADDTBL('N1',NEQT1,'REAL','OUTPUT')
      NP=ADDTBL('P',NELEM*NINTGP,'REAL','OUTPUT')
      NC=ADDTBL('COORDM',NELEM*3,'REAL','OUTPUT')
      NXIM1=ADDTBL('XIM1',NEQT,'REAL','OUTPUT')
      NX1IM1=ADDTBL('X1IM1',NEQT1,'REAL','OUTPUT')
      NX=LOCTBL('X','OUTPUT')
      NX1=LOCTBL('X1','OUTPUT')
      NNCOORD=LOCTBL('COORD','OUTPUT')
      NNCOORD1=LOCTBL('COORD1','OUTPUT')
      NCONNEC=LOCTBL('CONNEC','OUTPUT')
      NCONNEC1=LOCTBL('CONNEC1','OUTPUT')
      NIDEQ=LOCTBL('IDEQ','OUTPUT')
      NIDEQ1=LOCTBL('IDEQ1','OUTPUT')
      NCONSTR=LOCTBL('CONSTR','OUTPUT')
      NCONSTR1=LOCTBL('CONSTR1','OUTPUT')
      NLMGLOB=LOCTBL('LMGLOB','OUTPUT')
      NLMGLOB1=LOCTBL('LMGLOB1','OUTPUT')
*
*  IMPRESSION DES VARIABLES PRIMAIRES (VITESSES ET CONFORMATION)
*
      CALL PRTSOL(IA(NX),IA(NNCOORD),IA(NIDEQ),IA(NCONSTR),
     &            IA(NX1),IA(NNCOORD1),IA(NIDEQ1),IA(NCONSTR1),
     &             IA(NXIM1),IA(NX1IM1))
*
*  CALCUL DES VARIABLES SECONDAIRES (PRESSION)
*
      IELEMG=0
      DO 50 IGR=1,NGROUP
        TYPE    =BLOC(IGR,1)
        NELGRP  =BLOC(IGR,2)
        NDP     =BLOC(IGR,3)
        INTGV   =BLOC(IGR,4)
        INTGP   =BLOC(IGR,5)
        MVISC   =BLOC(IGR,6)
        MCP     =BLOC(IGR,7)
        MCOND   =BLOC(IGR,8)
        RHO     =PROP(IGR,1)
        VISCOS  =PROP(IGR,2)
        CP      =PROP(IGR,3)
        COND    =PROP(IGR,4)
*
        CALL VARSEC(IELEMG,IA(NCONNEC),IA(NNCOORD),IA(NLMGLOB),IA(NC))
 50   CONTINUE
*
*  IMPRESSION DES VARIABLES SECONDAIRES (PRESSIONS)
*
      CALL PRINTP(IA(NP),IA(NC),NELEM,NINTGP)
*
*  CALCUL DE LA BIREFRINGENCE (DIFFERENCE CONTRAINTES NORMALES)
*
      IELEMG=0
      IELEMG1=0
      DO 60 IGR=1,NGROUP
        TYPE    =BLOC(IGR,1)
        NELGRP  =BLOC(IGR,2)
```

```
        NDP     =BLOC(IGR,3)
        INTGV   =BLOC(IGR,4)
        INTGP   =BLOC(IGR,5)
        MVISC   =BLOC(IGR,6)
        MCP     =BLOC(IGR,7)
        MCOND   =BLOC(IGR,8)
        RHO     =PROP(IGR,1)
        VISCOS  =PROP(IGR,2)
        CP      =PROP(IGR,3)
        COND    =PROP(IGR,4)
*
        CALL BIREF(IELEMG,IA(NCONNEC),IA(NNCOORD),IA(NLMGLOB),
     &             IA(NLMGLOB1),NDP1)

*
*  IMPRESSION DE LA DIFFERENCE CONTRAINTES NORMALES
*
        CALL PRINTN1(IELEMG1,IA(NN1),IA(NNCOORD1),NDP1,NELGRP,
     &             IA(NCONNEC1))
 60   CONTINUE
      RETURN
      END
*
*     ================================================================
      SUBROUTINE PRTSOL(X,COORD,IDEQ,CONSTR,X1,COORD1,IDEQ1,CONSTR1,
     &                  XIM1,X1IM1)
*     ================================================================
*
*  FONCTION : IMPRESSION DE LA SOLUTION POUR LES VARIABLES PRIMAIRES
*
      IMPLICIT NONE
      INTEGER MAXVAR
      PARAMETER (MAXVAR=4)
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      INTEGER IDIM,ITEMP,NCOORD
      COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
      INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
      COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                 NUMNP1,NNPE1,NDEP1
      INTEGER NFIXB, IFIX, NFIXB1, IFIX1
      COMMON /FIX/ NFIXB, IFIX(MAXVAR), NFIXB1, IFIX1(3)
      INTEGER LDOF,KDOF
      COMMON/ DOF/ LDOF(MAXVAR),KDOF(MAXVAR)
      INTEGER NEQT,MELDOF,ESPMAT
      COMMON /MATDIM/ NEQT,MELDOF,ESPMAT
      INTEGER NEQT1,MELDOF1,ESPMAT1
      COMMON /MATDIM1/ NEQT1,MELDOF1,ESPMAT1
      REAL*8 DE
      COMMON / DEBORAH / DE
      INTEGER NEWTON
      COMMON / PREMIERE / NEWTON
      INTEGER I,J,IEQ
      INTEGER IDEQ(NUMNP,NDEP)
      INTEGER IDEQ1(NUMNP1,NDEP1)
      REAL*8 X( NEQT ),COORD( NUMNP ,NCOORD),CONSTR( NFIXB )
      REAL*8 X1(NEQT1),COORD1(NUMNP1,NCOORD),CONSTR1(NFIXB1)
      REAL*8 SOLUT(MAXVAR),SOLUT1(3),COORDT(3)
*
      REAL*8 XIM1(NEQT),X1IM1(NEQT1),NORML2,NORMAX
      REAL*8 N2DELTAX,NMDELTAX,N2DELTAX1,NMDELTAX1
      CHARACTER*75 LIGNE
*
      IF(NEWTON.EQ.1) THEN
*
*     IMPRESSION DES NORMES DES DIFFERENCES ENTRE LES ESTIMES INITIAUX DE LA
```

```
*     SOLUTION ET LA SOLUTION CALCULEE
*
        REWIND(UNIT=3)
        READ(3,2005) LIGNE
        DO 500 I=1,NEQT
          READ(3,*) XIM1(I)
          XIM1(I)=X(I)-XIM1(I)
 500    CONTINUE
        DO 501 I=1,NEQT1
          READ(3,*) X1IM1(I)
          X1IM1(I)=X1(I)-X1IM1(I)
 501    CONTINUE
        N2DELTAX = NORML2(XIM1,  NEQT)
        NMDELTAX = NORMAX(XIM1,  NEQT)
        N2DELTAX1= NORML2(X1IM1, NEQT1)
        NMDELTAX1= NORMAX(X1IM1, NEQT1)
        WRITE(IPRINT,5000) NMDELTAX
        WRITE(IPRINT,5001) N2DELTAX
        WRITE(IPRINT,5002) NMDELTAX1
        WRITE(IPRINT,5003) N2DELTAX1
      ENDIF
*
*
*     IMPRESSION DE LA SOLUTION (VITESSES)
*
      WRITE(IPRINT,1000)
      WRITE(5,     2000) DE
      WRITE(5,     1050)
      DO 100 I=1,NUMNP
        DO 200 J=1,MAXVAR
          SOLUT(J)=0.0D0
          IF(LDOF(J).NE.0) THEN
            IEQ=IDEQ(I,LDOF(J))
            IF(IEQ.LT.0) THEN
              SOLUT(J)=CONSTR(-IEQ)
            ELSEIF (IEQ.EQ.0) THEN
              SOLUT(J)=0.0D0
            ELSE
              SOLUT(J)=X(IEQ)
            ENDIF
          ENDIF
 200    CONTINUE
        DO 300 J=1,3
          COORDT(J)=0.0D0
          IF(J.LE.NCOORD) COORDT(J)=COORD(I,J)
 300    CONTINUE
        WRITE(IPRINT,1100) I, (SOLUT(J),J=1,MAXVAR),
     &                        (COORDT(J),J=1,3)
        WRITE(5     ,1150) (COORDT(J),J=1,3), I,
     &                        (SOLUT(J),J=1,MAXVAR)

 100  CONTINUE
      WRITE(IPRINT,1200)
*
*     IMPRESSION DE LA SOLUTION (CONFORMATION)
*
      WRITE(IPRINT,1001)
      WRITE(6,     2000) DE
      WRITE(6,     1061)
      DO 101 I=1,NUMNP1
        DO 201 J=1,3
          SOLUT1(J)=0.0D0
          IEQ=IDEQ1(I,J)
          IF(IEQ.LT.0) THEN
            SOLUT1(J)=CONSTR1(-IEQ)
          ELSEIF (IEQ.EQ.0) THEN
```

```
            SOLUT1(J)=0.0D0
          ELSE
            SOLUT1(J)=X1(IEQ)
          ENDIF
 201    CONTINUE
        DO 301 J=1,3
          COORDT(J)=0.0D0
          IF(J.LE.NCOORD) COORDT(J)=COORD1(I,J)
 301    CONTINUE
        WRITE(IPRINT,1101) I, (SOLUT1(J),J=1,3),
     &                         (COORDT(J),J=1,3)
        WRITE(6     ,1161) (COORDT(J),J=1,3), I,
     &                     (SOLUT1(J),J=1,3)

 101  CONTINUE
      WRITE(IPRINT,1200)
*
*     SAUVER LA SOLUTION DANS LE FICHIER "estimesu" COMME ESTIME
*     INITIAL POUR UN CALCUL A UN NOMBRE DE DEBORAH PLUS ELEVE
*
      REWIND(UNIT=4)
      WRITE(4,2000) DE
      DO 350 J=1,NEQT
         WRITE(4,*) X(J)
 350  CONTINUE
      DO 400 J=1,NEQT1
         WRITE(4,*) X1(J)
 400  CONTINUE
*
 1000 FORMAT(//,T25,'SOLUTION DES VARIABLES PRIMAIRES'//'I=======I',
     &       41('='),'I',28('='),'I'/'I NOEUD I',T20,
     &       'VARIABLES DEPENDANTES',T51,'I',T60,'COORDONNEES',T80,'I'/
     &       'I=======I',41('='),'I',28('='),'I'/
     &       'I     # I',T19,'U',T29,'V',T39,'W',T49,'T I',T53,
     &       'COORD #1',T62,'COORD #2',T71,'COORD #3 I'/'I=======I',
     &       41('='),'I',28('='),'I')
 1100 FORMAT('I',I6,' I',4(F10.5),' I',3(F9.5),' I')
 1001 FORMAT(//,T25,'SOLUTION DES VARIABLES PRIMAIRES'//'I=======I',
     &       41('='),'I',28('='),'I'/'I NOEUD I',T20,
     &       'VARIABLES DEPENDANTES',T51,'I',T60,'COORDONNEES',T80,'I'/
     &       'I=======I',41('='),'I',28('='),'I'/
     &       'I     # I',T19,'CXX',T32,'CYY',T45,'CXY   I',T53,
     &       'COORD #1',T62,'COORD #2',T71,'COORD #3 I'/'I=======I',
     &       41('='),'I',28('='),'I')
 1101 FORMAT('I',I6,' I',3(F13.5),'  I',3(F9.5),' I')
 1050 FORMAT('   COORD#1 ',T13,'COORD#2',T22,'COORD#3',T35,'NO.',T42,
     &       'U  ',T51,' V       ',T61,' W       ',T71,' T')
 1150 FORMAT(' ',3(F9.5),'  ',I6,'  ',4(F10.5),'  ')
 1061 FORMAT('   COORD#1 ',T13,'COORD#2',T22,'COORD#3',T35,'NO.',T42,
     &       '    CXX  ',T59,'CYY',T72,'CXY')
 1161 FORMAT(' ',3(F9.5),'  ',I6,'   ',3(F13.5),'  ')
 1200 FORMAT('I=======I',41('='),'I',28('='),'I')
 2000 FORMAT(' NOMBRE DE DEBORAH = ',F8.4)
 2005 FORMAT(A75)
 5000 FORMAT(' NORME MAXIMALE    X(I)-X(I-1)    :',F9.5)
 5001 FORMAT(' NORME EUCLEDIENNE X(I)-X(I-1)    :',F9.5)
 5002 FORMAT(' NORME MAXIMALE    X1(I)-X1(I-1) :',F9.5)
 5003 FORMAT(' NORME EUCLEDIENNE X1(I)-X1(I-1) :',F9.5)
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE VARSEC(IELEMG,CONNEC,COORD,LMGLOB,COORDM)
*     ================================================================
*
```

```
*  FONCTION : CALCUL DU PROFIL DE PRESSION
*
      IMPLICIT CHARACTER (A-Z)
      INTEGER MBLANK
      PARAMETER (MBLANK=2500000)
      INTEGER IA
      COMMON /BLANK/ IA(MBLANK)
      INTEGER IDIM,ITEMP,NCOORD
      COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
      INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
      COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                 NUMNP1,NNPE1,NDEP1
      INTEGER NFIXB,IFIX,NFIXB1,IFIX1
      COMMON /FIX/ NFIXB,IFIX(4),NFIXB1,IFIX1(3)
      INTEGER TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      COMMON /BLOC/ TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
      REAL*8 RHO,VISCOS,CP,COND
      COMMON /PROP/ RHO,VISCOS,CP,COND
      INTEGER NEQT,MELDOF,ESPMAT
      COMMON /MATDIM/ NEQT,MELDOF,ESPMAT
      REAL*8 EPS
      COMMON /EPS/ EPS
*
      INTEGER NINTGP,NXGP,NWP,NSHP,NDLP,NDGP,NJAP,IELEM,
     &        NCONSTR,NOEUD,NRGP,NP,NX,ADDTBL,LOCTBL
      REAL*8 COORDE(9,2),COORDM(NELEM,3)
      INTEGER LM(36),I,J
*
      INTEGER IELEMG,CONNEC(NELEM,NDP),LMGLOB(NELEM,MELDOF)
      REAL*8 COORD(NUMNP,NCOORD)
*
      NINTGP=INTGP**NCOORD
      NXGP=ADDTBL('XGAUSP',NINTGP*NCOORD,'REAL','OUTPUT')
      NWP=ADDTBL('WP',NINTGP,'REAL','OUTPUT')
      NSHP=ADDTBL('SHAPEP',NINTGP*NDP,'REAL','OUTPUT')
      NDLP=ADDTBL('DSLOCP',NINTGP*NDP*NCOORD,'REAL','OUTPUT')
      NDGP=ADDTBL('DSGLBP',NINTGP*NDP*NCOORD,'REAL','OUTPUT')
      NJAP=ADDTBL('DJACP',NINTGP,'REAL','OUTPUT')
      NRGP=ADDTBL('RAYONP',NINTGP,'REAL','OUTPUT')
      NCONSTR=LOCTBL('CONSTR','OUTPUT')
      NP=LOCTBL('P','OUTPUT')
      NX=LOCTBL('X','OUTPUT')
*
      CALL GAUSS(INTGP,NCOORD,NINTGP,IA(NXGP),IA(NWP))
      CALL FBASE(NINTGP,NDP,NCOORD,IA(NXGP),IA(NSHP),
     &           IA(NDLP),TYPE)
      DO 100 IELEM=1,NELGRP
        IELEMG=IELEMG+1
        DO 200 I=1,NDP
          NOEUD=CONNEC(IELEMG,I)
          DO 200 J=1,NCOORD
            COORDE(I,J)=COORD(NOEUD,J)
200     CONTINUE
        CALL DRGLB(NINTGP,NDP,NCOORD,COORDE,
     &             IA(NDLP),IA(NDGP),IA(NJAP))
        CALL RAYON(NINTGP,COORDE,IA(NXGP),IA(NRGP))
        DO 300 I=1,MELDOF
          LM(I)=LMGLOB(IELEMG,I)
300     CONTINUE
        CALL PROFILP(IELEMG,IA(NSHP),IA(NDGP),LM,NINTGP,NDP,
     &               NEQT,NFIXB,MELDOF,IA(NX),IA(NCONSTR),IA(NRGP),
     &               IA(NP))
*
*  CALCUL DES COORDONNEES MOYENNES DE L'ELEMENT
*  (A MODIFIER ULTERIEUREMENT PAR LE CALCUL DU CENTROIDE)
*
```

```
        COORDM(IELEMG,1)=(COORDE(1,1)+COORDE(2,1)+COORDE(3,1)+
     &                    COORDE(4,1))/4.0D0
        COORDM(IELEMG,2)=(COORDE(1,2)+COORDE(2,2)+COORDE(3,2)+
     &                    COORDE(4,2))/4.0D0
        COORDM(IELEMG,3)=0.0D0
 100  CONTINUE
      CALL RMVTBL('RAYONP')
      CALL RMVTBL('DJACP')
      CALL RMVTBL('DSGLBP')
      CALL RMVTBL('DSLOCP')
      CALL RMVTBL('SHAPEP')
      CALL RMVTBL('WP')
      CALL RMVTBL('XGAUSP')
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE PROFILP(IELEMG,SHAPE,DSGLB,LM,NPGAUS,NDP,
     &                   NBREQ,NBRCNT,NBRDLE,X,CONTNT,R,P)
*     ================================================================
*
*  FONCTION : RECUPERATION DU PROFIL DE PRESSION A PARTIR
*             DU PROFIL DE VITESSE
*
      IMPLICIT NONE
      INTEGER IDIM,ITEMP,NCOORD
      COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
      INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
      COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                 NUMNP1,NNPE1,NDEP1
      REAL*8 EPS
      COMMON /EPS/ EPS
      REAL*8 DE
      COMMON / DEBORAH / DE
      INTEGER NPGAUS,NDP
      INTEGER NBREQ,NBRCNT,NBRDLE
      INTEGER LM(NBRDLE),I,J,K,POSI,IELEMG
      REAL*8 X(NBREQ),CONTNT(NBRCNT),RESULT(9,2)
      REAL*8 DSGLB(NPGAUS,NDP,NCOORD),SHAPE(NPGAUS,NDP)
      REAL*8 R(NPGAUS),P(NELEM,NPGAUS)
      REAL*8 DUDX,DVDY,V,AXI
*
*  TEST COORDONNEES CARTESIENNES OU CYLINDRIQUES
*
      IF(IDIM.EQ.1.OR.IDIM.EQ.2) THEN
        AXI=1.0D0
      ELSEIF(IDIM.EQ.0.OR.IDIM.EQ.3) THEN
        AXI=0.0D0
      ENDIF
*
*  RECUPERATION DU PROFIL DE VITESSE DE L'ELEMENT
*
      POSI=0
      DO 20 J=1,2
        DO 10 I=1,NDP
          POSI=POSI+1
          IF(LM(POSI).LT.0) THEN
            RESULT(I,J)=CONTNT(-LM(POSI))
          ELSEIF(LM(POSI).EQ.0) THEN
            RESULT(I,J)=0.0D0
          ELSEIF(LM(POSI).GT.0) THEN
            RESULT(I,J)=X(LM(POSI))
          ENDIF
 10     CONTINUE
 20   CONTINUE
```

```
*
*  CALCUL DE LA DIVERGENCE
*
      DO 100 K=1,NPGAUS
        DUDX=0.0D0
        DVDY=0.0D0
        V   =0.0D0
        DO 200 I=1,NDP
          DUDX=DUDX+RESULT(I,1)*DSGLB(K,I,1)
          DVDY=DVDY+RESULT(I,2)*DSGLB(K,I,2)
          V   =   V+RESULT(I,2)*SHAPE(K,I)
 200  CONTINUE
*
*  CALCUL DE LA PRESSION
*
      P(IELEMG,K)=-(1.0D0*DE/EPS)*(DUDX+DVDY+AXI*(V/R(K)))
 100  CONTINUE
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE PRINTP(P,COORDM,NELEM,NPGAUS)
*     ================================================================
*
*  FONCTION : IMPRESSION DU PROFIL DE PRESSION
*
*
      IMPLICIT NONE
      INTEGER IPRINT
      COMMON /DISKS/ IPRINT
      REAL*8 DE
      COMMON /DEBORAH/ DE
      INTEGER NELEM,NPGAUS
      REAL*8 P(NELEM,NPGAUS),COORDM(NELEM,3),PMOY
      INTEGER IELEM,K
*
      IF(NPGAUS.NE.9) THEN
        WRITE(IPRINT,1000)
        DO 10 IELEM=1,NELEM
          WRITE(IPRINT,1100) IELEM,(P(IELEM,K),K=1,NPGAUS),
     &                       (COORDM(IELEM,K),K=1,3)
10      CONTINUE
        WRITE(IPRINT,1200)
      ENDIF
*
      WRITE(7,*) ' NOMBRE DE DEBORAH =', DE
*
      IF(NPGAUS.EQ.9) THEN
        WRITE(7,1079)
        DO 19 IELEM=1,NELEM
          PMOY=0.0D0
          DO 29 K=1,NPGAUS
            PMOY=PMOY+P(IELEM,K)
 29       CONTINUE
          PMOY=PMOY/9.0D0
          WRITE(7,        1179) (COORDM(IELEM,K),K=1,3),
     &                          IELEM, PMOY
 19     CONTINUE
      ENDIF
*
      IF(NPGAUS.EQ.4) THEN
        WRITE(7,1074)
        DO 14 IELEM=1,NELEM
          PMOY=0.0D0
          DO 24 K=1,NPGAUS
```

```
            PMOY=PMOY+P(IELEM,K)
 24       CONTINUE
          PMOY=PMOY/4.0D0
          WRITE(7,     1174) (COORDM(IELEM,K),K=1,3),
     &                        IELEM,(P(IELEM,K),K=1,NPGAUS),PMOY
 14     CONTINUE
       ENDIF
*
       IF(NPGAUS.EQ.1) THEN
         WRITE(7,1071)
         DO 11 IELEM=1,NELEM
           WRITE(7,       1171) (COORDM(IELEM,K),K=1,3),
     &                          IELEM,(P(IELEM,K),K=1,NPGAUS)
 11      CONTINUE
       ENDIF
*

 1000 FORMAT(//,T25,'SOLUTION DES VARIABLES SECONDAIRES'//'I=======I',
     &       41('='),'I',28('='),'I'/'IELEMENTI',T20,
     &       'VARIABLES DEPENDANTES',T51,'I',T60,'COORDONNEES',T80,'I'/
     &       'I=======I',41('='),'I',28('='),'I'/
     &       'I     # I',T24,'PRESSION',T51,'I',T53,
     &       'COORD #1',T62,'COORD #2',T71,'COORD #3 I'/'I=======I',
     &       41('='),'I',28('='),'I')
 1100 FORMAT('I',I6,' I',4(E10.3),' I',3(F9.5),' I')
 1079 FORMAT('   COORD#1',T16,'COORD#2',T24,'COORD#3',T32,
     &       'ELEMENT',T40,'PMOY')
 1179 FORMAT(' ' 3(F9.5),I6,2X,E10.4)
 1074 FORMAT(' COORD1',T9,'COORD2',T16,'COORD3',T24,
     &       ' ELEM',T31,'P1',T41,'P2',T51,'P3',T61,'P4',T70,'PMOY')
 1174 FORMAT(3(F7.4),I4,1X,5(E10.2))
 1071 FORMAT('  COORD#1',T10,'COORD#2',T20,'COORD#3',T30,
     &       'ELEMENT',T40,'P1')
 1171 FORMAT(' ' 3(F9.5),I6,2X,E10.3)

 1200 FORMAT('I=======I',41('='),'I',28('='),'I')
       RETURN
       END
*
*      ================================================================
       SUBROUTINE BIREF(IELEMG,CONNEC,COORD,LMGLOB,LMGLOB1,NDP1)
*      ================================================================
*
*  FONCTION : CALCUL DE LA BIREFRINGENCE
*
       IMPLICIT NONE
       INTEGER MBLANK
       PARAMETER (MBLANK=2500000)
       INTEGER IA
       COMMON /BLANK/ IA(MBLANK)
       INTEGER IDIM,ITEMP,NCOORD
       COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
       INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
       COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                  NUMNP1,NNPE1,NDEP1
       INTEGER NFIXB,IFIX,NFIXB1,IFIX1
       COMMON /FIX/ NFIXB,IFIX(4),NFIXB1,IFIX1(3)
       INTEGER TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
       COMMON /BLOC/ TYPE,NELGRP,NDP,INTGV,INTGP,MVISC,MCP,MCOND
       REAL*8 RHO,VISCOS,CP,COND
       COMMON /PROP/ RHO,VISCOS,CP,COND
       INTEGER NEQT,MELDOF,ESPMAT
       COMMON /MATDIM/ NEQT,MELDOF,ESPMAT
       INTEGER NEQT1,MELDOF1,ESPMAT1
       COMMON /MATDIM1/ NEQT1,MELDOF1,ESPMAT1
```

```
*
      INTEGER NINTGU,NXGU,NWU,NSHU,NDLU,NDGU,NJAU,IELEM,NDP1,
     &        NCONSTR,NCONSTR1,NOEUD,NX,NX1,NN1,ADDTBL,LOCTBL
      REAL*8 COORDE(9,2)
      INTEGER LM(36),LM1(12),I,J
*
      INTEGER IELEMG,CONNEC(NELEM,NDP),LMGLOB(NELEM,MELDOF)
      INTEGER LMGLOB1(NELEM,MELDOF1)
      REAL*8 COORD(NUMNP,NCOORD)
*
      NINTGU=INTGV**NCOORD
      NXGU=ADDTBL('XGAUSU',NINTGU*NCOORD,'REAL','OUTPUT')
      NWU=ADDTBL('WU',NINTGU,'REAL','OUTPUT')
      NSHU=ADDTBL('SHAPEU',NINTGU*NDP,'REAL','OUTPUT')
      NDLU=ADDTBL('DSLOCU',NINTGU*NDP*NCOORD,'REAL','OUTPUT')
      NDGU=ADDTBL('DSGLBU',NINTGU*NDP*NCOORD,'REAL','OUTPUT')
      NJAU=ADDTBL('DJACU',NINTGU,'REAL','OUTPUT')
      NCONSTR=LOCTBL('CONSTR','OUTPUT')
      NCONSTR1=LOCTBL('CONSTR1','OUTPUT')
      NN1=LOCTBL('N1','OUTPUT')
      NX=LOCTBL('X','OUTPUT')
      NX1=LOCTBL('X1','OUTPUT')
*
      CALL GAUSS(INTGV,NCOORD,NINTGU,IA(NXGU),IA(NWU))
      CALL FBASE(NINTGU,NDP,NCOORD,IA(NXGU),IA(NSHU),
     &           IA(NDLU),TYPE)
      DO 100 IELEM=1,NELGRP
        IELEMG=IELEMG+1
        DO 200 I=1,NDP
          NOEUD=CONNEC(IELEMG,I)
          DO 200 J=1,NCOORD
            COORDE(I,J)=COORD(NOEUD,J)
 200    CONTINUE
        CALL DRGLB(NINTGU,NDP,NCOORD,COORDE,
     &             IA(NDLU),IA(NDGU),IA(NJAU))
        DO 300 I=1,MELDOF
          LM(I)=LMGLOB(IELEMG,I)
 300    CONTINUE
        DO 400 I=1,MELDOF1
          LM1(I)=LMGLOB1(IELEMG,I)
 400    CONTINUE
        CALL CONTRAINTE(IELEMG,IA(NWU),IA(NDGU),LM,LM1,NINTGU,NDP,
     &                  NDP1,IA(NX),IA(NX1),IA(NCONSTR),IA(NCONSTR1),
     &                  IA(NN1))
*
 100  CONTINUE
      CALL RMVTBL('DJACU')
      CALL RMVTBL('DSGLBU')
      CALL RMVTBL('DSLOCU')
      CALL RMVTBL('SHAPEU')
      CALL RMVTBL('WU')
      CALL RMVTBL('XGAUSU')
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE CONTRAINTE(IELEMG,POIDS,DSGLB,LM,LM1,NPGAUS,NDP,
     &                      NDP1,X,X1,CONSTR,CONSTR1,N1)
*     ================================================================
*
*  FONCTION : CALCUL DE LA DIFFERENCE DES CONTRAINTES NORNALES A PARTIR
*             DU PROFIL DE VITESSES ET DU PROFIL DE CONFORMATION
*
      IMPLICIT NONE
      INTEGER IDIM,ITEMP,NCOORD
```

```
      COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
      INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
      COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                 NUMNP1,NNPE1,NDEP1
      INTEGER NEQT,MELDOF,ESPMAT
      COMMON /MATDIM/ NEQT,MELDOF,ESPMAT
      INTEGER NEQT1,MELDOF1,ESPMAT1
      COMMON /MATDIM1/ NEQT1,MELDOF1,ESPMAT1
      INTEGER NFIXB,IFIX,NFIXB1,IFIX1
      COMMON /FIX/ NFIXB,IFIX(4),NFIXB1,IFIX1(3)
      REAL*8 DE
      COMMON /DEBORAH/ DE
      INTEGER NPGAUS,NDP,NDP1
      INTEGER LM(MELDOF),LM1(MELDOF1),I,J,K,POSI,IELEMG
      REAL*8 X(NEQT),CONSTR(NFIXB),RESULT(9,2)
      REAL*8 X1(NEQT1),CONSTR1(NFIXB1),RESULT1(4,3)
      REAL*8 DSGLB(NPGAUS,NDP,NCOORD),POIDS(NPGAUS)
      REAL*8 N1(NELEM,NDP1)
      REAL*8 DUDX,DUDY,DVDX,DVDY,SOM,PARENT,SXX,SYY
*
*  RECUPERATION DU PROFIL DE VITESSE DE L'ELEMENT
*
      POSI=0
      DO 20 J=1,2
        DO 10 I=1,NDP
          POSI=POSI+1
          IF(LM(POSI).LT.0) THEN
            RESULT(I,J)=CONSTR(-LM(POSI))
          ELSEIF(LM(POSI).EQ.0) THEN
            RESULT(I,J)=0.0D0
          ELSEIF(LM(POSI).GT.0) THEN
            RESULT(I,J)=X(LM(POSI))
          ENDIF
 10     CONTINUE
 20   CONTINUE
*
*  RECUPERATION DU PROFIL DE CONFORMATION DE L'ELEMENT
*
      POSI=0
      DO 21 J=1,3
        DO 11 I=1,NDP1
          POSI=POSI+1
          IF(LM(POSI).LT.0) THEN
            RESULT1(I,J)=CONSTR1(-LM1(POSI))
          ELSEIF(LM1(POSI).EQ.0) THEN
            RESULT1(I,J)=0.0D0
          ELSEIF(LM1(POSI).GT.0) THEN
            RESULT1(I,J)=X1(LM1(POSI))
          ENDIF
 11     CONTINUE
 21   CONTINUE

*
*  CALCUL DES GRADIENTS DE VITESSES MOYENS DE L'ELEMENT
*
      DUDX=0.0D0
      DUDY=0.0D0
      DVDX=0.0D0
      DVDY=0.0D0
      SOM =0.0D0
      DO 100 K=1,NPGAUS
        SOM=SOM+POIDS(K)
        DO 200 I=1,NDP
          DUDX=DUDX+POIDS(K)*RESULT(I,1)*DSGLB(K,I,1)
          DUDY=DUDY+POIDS(K)*RESULT(I,1)*DSGLB(K,I,2)
          DVDX=DVDX+POIDS(K)*RESULT(I,2)*DSGLB(K,I,1)
```

```
          DVDY=DVDY+POIDS(K)*RESULT(I,2)*DSGLB(K,I,2)
 200    CONTINUE
 100  CONTINUE
      DUDX=DUDX/SOM
      DUDY=DUDY/SOM
      DVDX=DVDX/SOM
      DVDY=DVDY/SOM
*
*  CALCUL DE LA DIFFERENCE CONTRAINTES NORMALES PRIMAIRES
*
      DO 300 I=1,NDP1
        PARENT=((DUDY+DVDX)*RESULT1(I,3)+DUDX*RESULT1(I,1) +
     &           DVDY*RESULT1(I,2))
*
*     CALCUL DES COMPOSANTES DU TENSEUR DES EXTRA-CONTRAINTES
*
        SXX=1.0D0-3.0D0*RESULT1(I,1)-2.0D0*DE*PARENT*RESULT1(I,1)
        SYY=1.0D0-3.0D0*RESULT1(I,2)-2.0D0*DE*PARENT*RESULT1(I,2)
        N1(IELEMG,I) = -(SXX - SYY)
 300  CONTINUE
      RETURN
      END
*
*
*     ================================================================
      SUBROUTINE PRINTN1(IELEMG,N1,COORD1,NDP1, NELGRP,CONNEC1)
*     ================================================================
*
*     FONCTION: IMPRESSION DE LA DIFFERENCE DES CONTRAINTES NORMALES
*               PRIMAIRES
*
      IMPLICIT NONE
      INTEGER IDIM, ITEMP, NCOORD
      COMMON /CONTRL1/ IDIM,ITEMP,NCOORD
      INTEGER NUMNP,NELEM,NGROUP,NNPE,NDEP,NUMNP1,NNPE1,NDEP1
      COMMON /CONTRL2/ NUMNP,NELEM,NGROUP,NNPE,NDEP,
     &                 NUMNP1,NNPE1,NDEP1
      REAL*8 DE
      COMMON /DEBORAH/ DE
      INTEGER NDP1, NELGRP
      INTEGER CONNEC1(NELEM,NDP1)
      INTEGER I, J, IELEMG, NOEUD, IELEM
      REAL*8 N1(NELEM,NDP1), COORD1(NUMNP1,NCOORD)
      REAL*8 COORDE1(4,2),COORD3
*
      COORD3=0.0D0
      IF(IELEMG.EQ.0) THEN
        WRITE(8,     2000) DE
        WRITE(8,     1061)
      ENDIF
*
*  BOUCLE SUR LES ELEMENTS DU GROUPE
        DO 100 IELEM=1,NELGRP
          IELEMG=IELEMG+1
*
*  CALCUL DES COORDONNEES REELLES DE L'ELEMENT
*
          DO 201 I=1,NDP1
            NOEUD=CONNEC1(IELEMG,I)
            DO 101 J=1,NCOORD
              COORDE1(I,J)=COORD1(NOEUD,J)
 101        CONTINUE
*
*     IMPRESSION DE LA DIFFERENCE DES CONTRAINTES NORMALES  PRIMAIRES
*
            WRITE(8     ,1161) (COORDE1(I,J),J=1,2), COORD3, NOEUD, N1(IELEMG,I)
```

```
 201      CONTINUE
 100    CONTINUE
*
 1061 FORMAT('   COORD#1 ',T13,'COORD#2',T22,'COORD#3',T35,'NO.',T42,
     &       '    N1  ')
 1161 FORMAT(' ',3(F9.5),'  ',I6,'   ',(F13.5),'  ')
 2000 FORMAT(' NOMBRE DE DEBORAH = ',F8.4)
 2005 FORMAT(A75)
      RETURN
      END
```

sans l'impression de la interference ni de la différence des contraintes normales

EF2D: ECOULEMENT PLAN 2H=4 VITESSES => QUAD9, CONFORMATION => QUAD4

IDIM = 0 ITEMP = 0
NUMNP =153 NELEM = 32 NGROUP= 1 NNPE = 9
NFIXU = 16 NFIXV = 0 NFIXW = 0 NFIXT = 0
EPS =.1E-05 MAXITER=2000

NUMNP1 = 45
NFIXCXX = 9 NFIXCYY = 9 NFIXCXY = 8
MAXITER1= 30 MAXITGL = 20 UPWIND = 1

*** LISTBL *** APPELE DE LA ROUTINE : " INDATA"

LISTE DE TOUS LES TABLEAUX ALLOUES A CET INSTANT

NO.	TABLE	LONGUEUR	TYPE	DEBUT A	FIN A	ORIGINE
1 -	COORD	306	REAL	1	612	INDATA
2 -	COORD1	90	REAL	613	792	INDATA
3 -	IDEQ	306	INTEGER	793	1098	INDATA
4 -	IDEQ1	135	INTEGER	1099	1234	INDATA
5 -	CONSTR	16	REAL	1235	1266	INDATA
6 -	CONSTR1	26	REAL	1267	1318	INDATA
7 -	ID	306	INTEGER	1319	1624	INDATA
8 -	ID1	135	INTEGER	1625	1760	INDATA

NOMBRE DE DEBORAH = 0.2012

NOD	ID(I,J)				IDEQ(I,J)				COORDONNEES		
#	U	V	W	T	U	V	W	T	COORD #1	COORD #2	COORD #3
1	2	1	0	0	-1	0	0	0	0.00000	0.00000	0.00000
2	2	1	0	0	-2	0	0	0	0.00000	0.06250	0.00000
3	2	1	0	0	-3	0	0	0	0.00000	0.12500	0.00000
4	2	1	0	0	-4	0	0	0	0.00000	0.18750	0.00000
5	2	1	0	0	-5	0	0	0	0.00000	0.25000	0.00000
6	2	1	0	0	-6	0	0	0	0.00000	0.31250	0.00000
7	2	1	0	0	-7	0	0	0	0.00000	0.37500	0.00000
8	2	1	0	0	-8	0	0	0	0.00000	0.43750	0.00000
9	2	1	0	0	-9	0	0	0	0.00000	0.50000	0.00000
10	2	1	0	0	-10	0	0	0	0.00000	0.56250	0.00000
11	2	1	0	0	-11	0	0	0	0.00000	0.62500	0.00000
12	2	1	0	0	-12	0	0	0	0.00000	0.68750	0.00000
13	2	1	0	0	-13	0	0	0	0.00000	0.75000	0.00000
14	2	1	0	0	-14	0	0	0	0.00000	0.81250	0.00000
15	2	1	0	0	-15	0	0	0	0.00000	0.87500	0.00000
16	2	1	0	0	-16	0	0	0	0.00000	0.93750	0.00000
17	1	1	0	0	0	0	0	0	0.00000	1.00000	0.00000
18	0	1	0	0	1	0	0	0	0.25000	0.00000	0.00000
19	0	0	0	0	2	3	0	0	0.25000	0.06250	0.00000
20	0	0	0	0	4	5	0	0	0.25000	0.12500	0.00000
21	0	0	0	0	6	7	0	0	0.25000	0.18750	0.00000
22	0	0	0	0	8	9	0	0	0.25000	0.25000	0.00000
23	0	0	0	0	10	11	0	0	0.25000	0.31250	0.00000
24	0	0	0	0	12	13	0	0	0.25000	0.37500	0.00000
25	0	0	0	0	14	15	0	0	0.25000	0.43750	0.00000
26	0	0	0	0	16	17	0	0	0.25000	0.50000	0.00000
27	0	0	0	0	18	19	0	0	0.25000	0.56250	0.00000

I	28	I	0	0	0	0 I	20	21	0	0 I	0.25000	0.62500	0.00000 I
I	29	I	0	0	0	0 I	22	23	0	0 I	0.25000	0.68750	0.00000 I
I	30	I	0	0	0	0 I	24	25	0	0 I	0.25000	0.75000	0.00000 I
I	31	I	0	0	0	0 I	26	27	0	0 I	0.25000	0.81250	0.00000 I
I	32	I	0	0	0	0 I	28	29	0	0 I	0.25000	0.87500	0.00000 I
I	33	I	0	0	0	0 I	30	31	0	0 I	0.25000	0.93750	0.00000 I
I	34	I	1	1	0	0 I	0	0	0	0 I	0.25000	1.00000	0.00000 I
I	35	I	0	1	0	0 I	32	0	0	0 I	0.50000	0.00000	0.00000 I
I	36	I	0	0	0	0 I	33	34	0	0 I	0.50000	0.06250	0.00000 I
I	37	I	0	0	0	0 I	35	36	0	0 I	0.50000	0.12500	0.00000 I
I	38	I	0	0	0	0 I	37	38	0	0 I	0.50000	0.18750	0.00000 I
I	39	I	0	0	0	0 I	39	40	0	0 I	0.50000	0.25000	0.00000 I
I	40	I	0	0	0	0 I	41	42	0	0 I	0.50000	0.31250	0.00000 I
I	41	I	0	0	0	0 I	43	44	0	0 I	0.50000	0.37500	0.00000 I
I	42	I	0	0	0	0 I	45	46	0	0 I	0.50000	0.43750	0.00000 I
I	43	I	0	0	0	0 I	47	48	0	0 I	0.50000	0.50000	0.00000 I
I	44	I	0	0	0	0 I	49	50	0	0 I	0.50000	0.56250	0.00000 I
I	45	I	0	0	0	0 I	51	52	0	0 I	0.50000	0.62500	0.00000 I
I	46	I	0	0	0	0 I	53	54	0	0 I	0.50000	0.68750	0.00000 I
I	47	I	0	0	0	0 I	55	56	0	0 I	0.50000	0.75000	0.00000 I
I	48	I	0	0	0	0 I	57	58	0	0 I	0.50000	0.81250	0.00000 I
I	49	I	0	0	0	0 I	59	60	0	0 I	0.50000	0.87500	0.00000 I
I	50	I	0	0	0	0 I	61	62	0	0 I	0.50000	0.93750	0.00000 I
I	51	I	1	1	0	0 I	0	0	0	0 I	0.50000	1.00000	0.00000 I
I	52	I	0	1	0	0 I	63	0	0	0 I	0.75000	0.00000	0.00000 I
I	53	I	0	0	0	0 I	64	65	0	0 I	0.75000	0.06250	0.00000 I
I	54	I	0	0	0	0 I	66	67	0	0 I	0.75000	0.12500	0.00000 I
I	55	I	0	0	0	0 I	68	69	0	0 I	0.75000	0.18750	0.00000 I
I	56	I	0	0	0	0 I	70	71	0	0 I	0.75000	0.25000	0.00000 I
I	57	I	0	0	0	0 I	72	73	0	0 I	0.75000	0.31250	0.00000 I
I	58	I	0	0	0	0 I	74	75	0	0 I	0.75000	0.37500	0.00000 I
I	59	I	0	0	0	0 I	76	77	0	0 I	0.75000	0.43750	0.00000 I
I	60	I	0	0	0	0 I	78	79	0	0 I	0.75000	0.50000	0.00000 I
I	61	I	0	0	0	0 I	80	81	0	0 I	0.75000	0.56250	0.00000 I
I	62	I	0	0	0	0 I	82	83	0	0 I	0.75000	0.62500	0.00000 I
I	63	I	0	0	0	0 I	84	85	0	0 I	0.75000	0.68750	0.00000 I
I	64	I	0	0	0	0 I	86	87	0	0 I	0.75000	0.75000	0.00000 I
I	65	I	0	0	0	0 I	88	89	0	0 I	0.75000	0.81250	0.00000 I
I	66	I	0	0	0	0 I	90	91	0	0 I	0.75000	0.87500	0.00000 I
I	67	I	0	0	0	0 I	92	93	0	0 I	0.75000	0.93750	0.00000 I
I	68	I	1	1	0	0 I	0	0	0	0 I	0.75000	1.00000	0.00000 I
I	69	I	0	1	0	0 I	94	0	0	0 I	1.00000	0.00000	0.00000 I
I	70	I	0	0	0	0 I	95	96	0	0 I	1.00000	0.06250	0.00000 I
I	71	I	0	0	0	0 I	97	98	0	0 I	1.00000	0.12500	0.00000 I
I	72	I	0	0	0	0 I	99	100	0	0 I	1.00000	0.18750	0.00000 I
I	73	I	0	0	0	0 I	101	102	0	0 I	1.00000	0.25000	0.00000 I
I	74	I	0	0	0	0 I	103	104	0	0 I	1.00000	0.31250	0.00000 I
I	75	I	0	0	0	0 I	105	106	0	0 I	1.00000	0.37500	0.00000 I
I	76	I	0	0	0	0 I	107	108	0	0 I	1.00000	0.43750	0.00000 I
I	77	I	0	0	0	0 I	109	110	0	0 I	1.00000	0.50000	0.00000 I
I	78	I	0	0	0	0 I	111	112	0	0 I	1.00000	0.56250	0.00000 I
I	79	I	0	0	0	0 I	113	114	0	0 I	1.00000	0.62500	0.00000 I
I	80	I	0	0	0	0 I	115	116	0	0 I	1.00000	0.68750	0.00000 I
I	81	I	0	0	0	0 I	117	118	0	0 I	1.00000	0.75000	0.00000 I
I	82	I	0	0	0	0 I	119	120	0	0 I	1.00000	0.81250	0.00000 I
I	83	I	0	0	0	0 I	121	122	0	0 I	1.00000	0.87500	0.00000 I
I	84	I	0	0	0	0 I	123	124	0	0 I	1.00000	0.93750	0.00000 I
I	85	I	1	1	0	0 I	0	0	0	0 I	1.00000	1.00000	0.00000 I
I	86	I	0	1	0	0 I	125	0	0	0 I	1.12500	0.00000	0.00000 I
I	87	I	0	0	0	0 I	126	127	0	0 I	1.12500	0.06250	0.00000 I
I	88	I	0	0	0	0 I	128	129	0	0 I	1.12500	0.12500	0.00000 I
I	89	I	0	0	0	0 I	130	131	0	0 I	1.12500	0.18750	0.00000 I
I	90	I	0	0	0	0 I	132	133	0	0 I	1.12500	0.25000	0.00000 I
I	91	I	0	0	0	0 I	134	135	0	0 I	1.12500	0.31250	0.00000 I
I	92	I	0	0	0	0 I	136	137	0	0 I	1.12500	0.37500	0.00000 I
I	93	I	0	0	0	0 I	138	139	0	0 I	1.12500	0.43750	0.00000 I

94	0	0	0	0	140	141	0	0	1.12500	0.50000	0.00000
95	0	0	0	0	142	143	0	0	1.12500	0.56250	0.00000
96	0	0	0	0	144	145	0	0	1.12500	0.62500	0.00000
97	0	0	0	0	146	147	0	0	1.12500	0.68750	0.00000
98	0	0	0	0	148	149	0	0	1.12500	0.75000	0.00000
99	0	0	0	0	150	151	0	0	1.12500	0.81250	0.00000
100	0	0	0	0	152	153	0	0	1.12500	0.87500	0.00000
101	0	0	0	0	154	155	0	0	1.12500	0.93750	0.00000
102	1	1	0	0	0	0	0	0	1.12500	1.00000	0.00000
103	0	1	0	0	156	0	0	0	1.25000	0.00000	0.00000
104	0	0	0	0	157	158	0	0	1.25000	0.06250	0.00000
105	0	0	0	0	159	160	0	0	1.25000	0.12500	0.00000
106	0	0	0	0	161	162	0	0	1.25000	0.18750	0.00000
107	0	0	0	0	163	164	0	0	1.25000	0.25000	0.00000
108	0	0	0	0	165	166	0	0	1.25000	0.31250	0.00000
109	0	0	0	0	167	168	0	0	1.25000	0.37500	0.00000
110	0	0	0	0	169	170	0	0	1.25000	0.43750	0.00000
111	0	0	0	0	171	172	0	0	1.25000	0.50000	0.00000
112	0	0	0	0	173	174	0	0	1.25000	0.56250	0.00000
113	0	0	0	0	175	176	0	0	1.25000	0.62500	0.00000
114	0	0	0	0	177	178	0	0	1.25000	0.68750	0.00000
115	0	0	0	0	179	180	0	0	1.25000	0.75000	0.00000
116	0	0	0	0	181	182	0	0	1.25000	0.81250	0.00000
117	0	0	0	0	183	184	0	0	1.25000	0.87500	0.00000
118	0	0	0	0	185	186	0	0	1.25000	0.93750	0.00000
119	1	1	0	0	0	0	0	0	1.25000	1.00000	0.00000
120	0	1	0	0	187	0	0	0	1.31250	0.00000	0.00000
121	0	0	0	0	188	189	0	0	1.31250	0.06250	0.00000
122	0	0	0	0	190	191	0	0	1.31250	0.12500	0.00000
123	0	0	0	0	192	193	0	0	1.31250	0.18750	0.00000
124	0	0	0	0	194	195	0	0	1.31250	0.25000	0.00000
125	0	0	0	0	196	197	0	0	1.31250	0.31250	0.00000
126	0	0	0	0	198	199	0	0	1.31250	0.37500	0.00000
127	0	0	0	0	200	201	0	0	1.31250	0.43750	0.00000
128	0	0	0	0	202	203	0	0	1.31250	0.50000	0.00000
129	0	0	0	0	204	205	0	0	1.31250	0.56250	0.00000
130	0	0	0	0	206	207	0	0	1.31250	0.62500	0.00000
131	0	0	0	0	208	209	0	0	1.31250	0.68750	0.00000
132	0	0	0	0	210	211	0	0	1.31250	0.75000	0.00000
133	0	0	0	0	212	213	0	0	1.31250	0.81250	0.00000
134	0	0	0	0	214	215	0	0	1.31250	0.87500	0.00000
135	0	0	0	0	216	217	0	0	1.31250	0.93750	0.00000
136	1	1	0	0	0	0	0	0	1.31250	1.00000	0.00000
137	0	1	0	0	218	0	0	0	1.37500	0.00000	0.00000
138	0	1	0	0	219	0	0	0	1.37500	0.06250	0.00000
139	0	1	0	0	220	0	0	0	1.37500	0.12500	0.00000
140	0	1	0	0	221	0	0	0	1.37500	0.18750	0.00000
141	0	1	0	0	222	0	0	0	1.37500	0.25000	0.00000
142	0	1	0	0	223	0	0	0	1.37500	0.31250	0.00000
143	0	1	0	0	224	0	0	0	1.37500	0.37500	0.00000
144	0	1	0	0	225	0	0	0	1.37500	0.43750	0.00000
145	0	1	0	0	226	0	0	0	1.37500	0.50000	0.00000
146	0	1	0	0	227	0	0	0	1.37500	0.56250	0.00000
147	0	1	0	0	228	0	0	0	1.37500	0.62500	0.00000
148	0	1	0	0	229	0	0	0	1.37500	0.68750	0.00000
149	0	1	0	0	230	0	0	0	1.37500	0.75000	0.00000
150	0	1	0	0	231	0	0	0	1.37500	0.81250	0.00000
151	0	1	0	0	232	0	0	0	1.37500	0.87500	0.00000
152	0	1	0	0	233	0	0	0	1.37500	0.93750	0.00000
153	1	1	0	0	0	0	0	0	1.37500	1.00000	0.00000

NOD	ID1(I,J)	IDEQ1(I,J)	COORDONNEES

#	CXX	CYY	CXY	CXX	CYY	CXY	COORD #1	COORD #2	COORD #3
1	2	2	1	-1	-2	0	0.00000	0.00000	0.00000
2	2	2	2	-3	-4	-5	0.00000	0.12500	0.00000
3	2	2	2	-6	-7	-8	0.00000	0.25000	0.00000
4	2	2	2	-9	-10	-11	0.00000	0.37500	0.00000
5	2	2	2	-12	-13	-14	0.00000	0.50000	0.00000
6	2	2	2	-15	-16	-17	0.00000	0.62500	0.00000
7	2	2	2	-18	-19	-20	0.00000	0.75000	0.00000
8	2	2	2	-21	-22	-23	0.00000	0.87500	0.00000
9	2	2	2	-24	-25	-26	0.00000	1.00000	0.00000
10	0	0	0	1	2	3	0.50000	0.00000	0.00000
11	0	0	0	4	5	6	0.50000	0.12500	0.00000
12	0	0	0	7	8	9	0.50000	0.25000	0.00000
13	0	0	0	10	11	12	0.50000	0.37500	0.00000
14	0	0	0	13	14	15	0.50000	0.50000	0.00000
15	0	0	0	16	17	18	0.50000	0.62500	0.00000
16	0	0	0	19	20	21	0.50000	0.75000	0.00000
17	0	0	0	22	23	24	0.50000	0.87500	0.00000
18	0	0	0	25	26	27	0.50000	1.00000	0.00000
19	0	0	0	28	29	30	1.00000	0.00000	0.00000
20	0	0	0	31	32	33	1.00000	0.12500	0.00000
21	0	0	0	34	35	36	1.00000	0.25000	0.00000
22	0	0	0	37	38	39	1.00000	0.37500	0.00000
23	0	0	0	40	41	42	1.00000	0.50000	0.00000
24	0	0	0	43	44	45	1.00000	0.62500	0.00000
25	0	0	0	46	47	48	1.00000	0.75000	0.00000
26	0	0	0	49	50	51	1.00000	0.87500	0.00000
27	0	0	0	52	53	54	1.00000	1.00000	0.00000
28	0	0	0	55	56	57	1.25000	0.00000	0.00000
29	0	0	0	58	59	60	1.25000	0.12500	0.00000
30	0	0	0	61	62	63	1.25000	0.25000	0.00000
31	0	0	0	64	65	66	1.25000	0.37500	0.00000
32	0	0	0	67	68	69	1.25000	0.50000	0.00000
33	0	0	0	70	71	72	1.25000	0.62500	0.00000
34	0	0	0	73	74	75	1.25000	0.75000	0.00000
35	0	0	0	76	77	78	1.25000	0.87500	0.00000
36	0	0	0	79	80	81	1.25000	1.00000	0.00000
37	0	0	0	82	83	84	1.37500	0.00000	0.00000
38	0	0	0	85	86	87	1.37500	0.12500	0.00000
39	0	0	0	88	89	90	1.37500	0.25000	0.00000
40	0	0	0	91	92	93	1.37500	0.37500	0.00000
41	0	0	0	94	95	96	1.37500	0.50000	0.00000
42	0	0	0	97	98	99	1.37500	0.62500	0.00000
43	0	0	0	100	101	102	1.37500	0.75000	0.00000
44	0	0	0	103	104	105	1.37500	0.87500	0.00000
45	0	0	0	106	107	108	1.37500	1.00000	0.00000

TABLEAU CONSTR(NFIXB)

NOEUD	NOM	VALEUR
1	U	1.000000
2	U	0.996117
3	U	0.984466
4	U	0.965044
5	U	0.937845
6	U	0.902863
7	U	0.860087
8	U	0.809506
9	U	0.751105
10	U	0.684868
11	U	0.610777
12	U	0.528812
13	U	0.438950

14	U	0.341165
15	U	0.235432
16	U	0.121721

TABLEAU CONSTR1(NFIXB1)

NOEUD	NOM	VALEUR
1	CXX	0.333333
1	CYY	0.333333
2	CXX	0.333457
2	CYY	0.333272
2	CXY	-0.005555
3	CXX	0.333827
3	CYY	0.333087
3	CXY	-0.011103
4	CXX	0.334443
4	CYY	0.332779
4	CXY	-0.016639
5	CXX	0.335303
5	CYY	0.332349
5	CXY	-0.022157
6	CXX	0.336406
6	CYY	0.331797
6	CXY	-0.027650
7	CXX	0.337748
7	CYY	0.331126
7	CXY	-0.033113
8	CXX	0.339328
8	CYY	0.330336
8	CXY	-0.038539
9	CXX	0.341142
9	CYY	0.329429
9	CXY	-0.043924

*** LISTBL *** APPELE DE LA ROUTINE : " INDATA"

LISTE DE TOUS LES TABLEAUX ALLOUES A CET INSTANT

NO.	TABLE	LONGUEUR	TYPE	DEBUT A	FIN A	ORIGINE
1 -	COORD	306	REAL	1	612	INDATA
2 -	COORD1	90	REAL	613	792	INDATA
3 -	IDEQ	306	INTEGER	793	1098	INDATA
4 -	IDEQ1	135	INTEGER	1099	1234	INDATA
5 -	CONSTR	16	REAL	1235	1266	INDATA
6 -	CONSTR1	26	REAL	1267	1318	INDATA
7 -	CONNEC	288	INTEGER	1319	1606	INDATA
8 -	CONNEC1	128	INTEGER	1607	1734	INDATA
9 -	LMGLOB	576	INTEGER	1735	2310	INDATA
10 -	LMGLOB1	384	INTEGER	2311	2694	INDATA
11 -	DIAG	233	INTEGER	2695	2928	INDATA
12 -	DIAG1	108	INTEGER	2929	3036	INDATA
13 -	LCIEL	233	INTEGER	3037	3270	INDATA
14 -	LCIEL1	108	INTEGER	3271	3378	INDATA

**GROUPE D'ELEMENTS NO. 1

TYPE = 2 NELGRP= 32 NDP = 9 INTGV = 3
INTGP = 2 MVISC = 0 MCP = 0 MCOND = 0

RHO = 1.0000
VISCOS= 0.3333

ESPACE MATRICE "A" = 18853

ESPACE MATRICE "A1"= 5274

TABLE DE CONNECTIVITE

ELEM #	NUMEROTATION LOCALE DES NOEUDS								
	1	2	3	4	5	6	7	8	9
1	1	35	37	3	18	36	20	2	19
2	35	69	71	37	52	70	54	36	53
3	3	37	39	5	20	38	22	4	21
4	37	71	73	39	54	72	56	38	55
5	5	39	41	7	22	40	24	6	23
6	39	73	75	41	56	74	58	40	57
7	7	41	43	9	24	42	26	8	25
8	41	75	77	43	58	76	60	42	59
9	9	43	45	11	26	44	28	10	27
10	43	77	79	45	60	78	62	44	61
11	11	45	47	13	28	46	30	12	29
12	45	79	81	47	62	80	64	46	63
13	13	47	49	15	30	48	32	14	31
14	47	81	83	49	64	82	66	48	65
15	15	49	51	17	32	50	34	16	33
16	49	83	85	51	66	84	68	50	67
17	69	103	105	71	86	104	88	70	87
18	71	105	107	73	88	106	90	72	89
19	73	107	109	75	90	108	92	74	91
20	75	109	111	77	92	110	94	76	93
21	77	111	113	79	94	112	96	78	95
22	79	113	115	81	96	114	98	80	97
23	81	115	117	83	98	116	100	82	99
24	83	117	119	85	100	118	102	84	101
25	103	137	139	105	120	138	122	104	121
26	105	139	141	107	122	140	124	106	123
27	107	141	143	109	124	142	126	108	125
28	109	143	145	111	126	144	128	110	127
29	111	145	147	113	128	146	130	112	129
30	113	147	149	115	130	148	132	114	131
31	115	149	151	117	132	150	134	116	133
32	117	151	153	119	134	152	136	118	135

STRUCTURE EN LIGNE DE CIEL

EQ.#	LCIEL(I)	DIAG(I)
1	0	1
2	1	4
3	2	9
4	3	16
5	4	25
6	2	30
7	3	37
8	4	46

I	9 I	5 I	57 I
I	10 I	2 I	62 I
I	11 I	3 I	69 I
I	12 I	4 I	78 I
I	13 I	5 I	89 I
I	14 I	2 I	94 I
I	15 I	3 I	101 I
I	16 I	4 I	110 I
I	17 I	5 I	121 I
I	18 I	2 I	126 I
I	19 I	3 I	133 I
I	20 I	4 I	142 I
I	21 I	5 I	153 I
I	22 I	2 I	158 I
I	23 I	3 I	165 I
I	24 I	4 I	174 I
I	25 I	5 I	185 I
I	26 I	2 I	190 I
I	27 I	3 I	197 I
I	28 I	4 I	206 I
I	29 I	5 I	217 I
I	30 I	2 I	222 I
I	31 I	3 I	229 I
I	32 I	31 I	292 I
I	33 I	32 I	357 I
I	34 I	33 I	424 I
I	35 I	34 I	493 I
I	36 I	35 I	564 I
I	37 I	33 I	631 I
I	38 I	34 I	700 I
I	39 I	35 I	771 I
I	40 I	36 I	844 I
I	41 I	33 I	911 I
I	42 I	34 I	980 I
I	43 I	35 I	1051 I
I	44 I	36 I	1124 I
I	45 I	33 I	1191 I
I	46 I	34 I	1260 I
I	47 I	35 I	1331 I
I	48 I	36 I	1404 I
I	49 I	33 I	1471 I
I	50 I	34 I	1540 I
I	51 I	35 I	1611 I
I	52 I	36 I	1684 I
I	53 I	33 I	1751 I
I	54 I	34 I	1820 I
I	55 I	35 I	1891 I
I	56 I	36 I	1964 I
I	57 I	33 I	2031 I
I	58 I	34 I	2100 I
I	59 I	35 I	2171 I
I	60 I	36 I	2244 I
I	61 I	33 I	2311 I
I	62 I	34 I	2380 I
I	63 I	31 I	2443 I
I	64 I	32 I	2508 I
I	65 I	33 I	2575 I
I	66 I	34 I	2644 I
I	67 I	35 I	2715 I
I	68 I	33 I	2782 I
I	69 I	34 I	2851 I
I	70 I	35 I	2922 I
I	71 I	36 I	2995 I
I	72 I	33 I	3062 I
I	73 I	34 I	3131 I
I	74 I	35 I	3202 I

I	75 I	36 I	3275 I
I	76 I	33 I	3342 I
I	77 I	34 I	3411 I
I	78 I	35 I	3482 I
I	79 I	36 I	3555 I
I	80 I	33 I	3622 I
I	81 I	34 I	3691 I
I	82 I	35 I	3762 I
I	83 I	36 I	3835 I
I	84 I	33 I	3902 I
I	85 I	34 I	3971 I
I	86 I	35 I	4042 I
I	87 I	36 I	4115 I
I	88 I	33 I	4182 I
I	89 I	34 I	4251 I
I	90 I	35 I	4322 I
I	91 I	36 I	4395 I
I	92 I	33 I	4462 I
I	93 I	34 I	4531 I
I	94 I	62 I	4656 I
I	95 I	63 I	4783 I
I	96 I	64 I	4912 I
I	97 I	65 I	5043 I
I	98 I	66 I	5176 I
I	99 I	64 I	5305 I
I	100 I	65 I	5436 I
I	101 I	66 I	5569 I
I	102 I	67 I	5704 I
I	103 I	64 I	5833 I
I	104 I	65 I	5964 I
I	105 I	66 I	6097 I
I	106 I	67 I	6232 I
I	107 I	64 I	6361 I
I	108 I	65 I	6492 I
I	109 I	66 I	6625 I
I	110 I	67 I	6760 I
I	111 I	64 I	6889 I
I	112 I	65 I	7020 I
I	113 I	66 I	7153 I
I	114 I	67 I	7288 I
I	115 I	64 I	7417 I
I	116 I	65 I	7548 I
I	117 I	66 I	7681 I
I	118 I	67 I	7816 I
I	119 I	64 I	7945 I
I	120 I	65 I	8076 I
I	121 I	66 I	8209 I
I	122 I	67 I	8344 I
I	123 I	64 I	8473 I
I	124 I	65 I	8604 I
I	125 I	31 I	8667 I
I	126 I	32 I	8732 I
I	127 I	33 I	8799 I
I	128 I	34 I	8868 I
I	129 I	35 I	8939 I
I	130 I	33 I	9006 I
I	131 I	34 I	9075 I
I	132 I	35 I	9146 I
I	133 I	36 I	9219 I
I	134 I	33 I	9286 I
I	135 I	34 I	9355 I
I	136 I	35 I	9426 I
I	137 I	36 I	9499 I
I	138 I	33 I	9566 I
I	139 I	34 I	9635 I
I	140 I	35 I	9706 I

I	141 I	36 I	9779 I
I	142 I	33 I	9846 I
I	143 I	34 I	9915 I
I	144 I	35 I	9986 I
I	145 I	36 I	10059 I
I	146 I	33 I	10126 I
I	147 I	34 I	10195 I
I	148 I	35 I	10266 I
I	149 I	36 I	10339 I
I	150 I	33 I	10406 I
I	151 I	34 I	10475 I
I	152 I	35 I	10546 I
I	153 I	36 I	10619 I
I	154 I	33 I	10686 I
I	155 I	34 I	10755 I
I	156 I	62 I	10880 I
I	157 I	63 I	11007 I
I	158 I	64 I	11136 I
I	159 I	65 I	11267 I
I	160 I	66 I	11400 I
I	161 I	64 I	11529 I
I	162 I	65 I	11660 I
I	163 I	66 I	11793 I
I	164 I	67 I	11928 I
I	165 I	64 I	12057 I
I	166 I	65 I	12188 I
I	167 I	66 I	12321 I
I	168 I	67 I	12456 I
I	169 I	64 I	12585 I
I	170 I	65 I	12716 I
I	171 I	66 I	12849 I
I	172 I	67 I	12984 I
I	173 I	64 I	13113 I
I	174 I	65 I	13244 I
I	175 I	66 I	13377 I
I	176 I	67 I	13512 I
I	177 I	64 I	13641 I
I	178 I	65 I	13772 I
I	179 I	66 I	13905 I
I	180 I	67 I	14040 I
I	181 I	64 I	14169 I
I	182 I	65 I	14300 I
I	183 I	66 I	14433 I
I	184 I	67 I	14568 I
I	185 I	64 I	14697 I
I	186 I	65 I	14828 I
I	187 I	31 I	14891 I
I	188 I	32 I	14956 I
I	189 I	33 I	15023 I
I	190 I	34 I	15092 I
I	191 I	35 I	15163 I
I	192 I	33 I	15230 I
I	193 I	34 I	15299 I
I	194 I	35 I	15370 I
I	195 I	36 I	15443 I
I	196 I	33 I	15510 I
I	197 I	34 I	15579 I
I	198 I	35 I	15650 I
I	199 I	36 I	15723 I
I	200 I	33 I	15790 I
I	201 I	34 I	15859 I
I	202 I	35 I	15930 I
I	203 I	36 I	16003 I
I	204 I	33 I	16070 I
I	205 I	34 I	16139 I
I	206 I	35 I	16210 I

207	36	16283
208	33	16350
209	34	16419
210	35	16490
211	36	16563
212	33	16630
213	34	16699
214	35	16770
215	36	16843
216	33	16910
217	34	16979
218	62	17104
219	63	17231
220	64	17360
221	62	17485
222	63	17612
223	60	17733
224	61	17856
225	58	17973
226	59	18092
227	56	18205
228	57	18320
229	54	18429
230	55	18540
231	52	18645
232	53	18752
233	50	18853

TABLE DE CONNECTIVITE

ELEM	NUMEROTATION LOCALE DES NOEUDS								
#	1	2	3	4	5	6	7	8	9
1	1	10	11	2	0	0	0	0	0
2	10	19	20	11	0	0	0	0	0
3	2	11	12	3	0	0	0	0	0
4	11	20	21	12	0	0	0	0	0
5	3	12	13	4	0	0	0	0	0
6	12	21	22	13	0	0	0	0	0
7	4	13	14	5	0	0	0	0	0
8	13	22	23	14	0	0	0	0	0
9	5	14	15	6	0	0	0	0	0
10	14	23	24	15	0	0	0	0	0
11	6	15	16	7	0	0	0	0	0
12	15	24	25	16	0	0	0	0	0
13	7	16	17	8	0	0	0	0	0
14	16	25	26	17	0	0	0	0	0
15	8	17	18	9	0	0	0	0	0
16	17	26	27	18	0	0	0	0	0
17	19	28	29	20	0	0	0	0	0
18	20	29	30	21	0	0	0	0	0
19	21	30	31	22	0	0	0	0	0
20	22	31	32	23	0	0	0	0	0
21	23	32	33	24	0	0	0	0	0
22	24	33	34	25	0	0	0	0	0
23	25	34	35	26	0	0	0	0	0
24	26	35	36	27	0	0	0	0	0
25	28	37	38	29	0	0	0	0	0
26	29	38	39	30	0	0	0	0	0
27	30	39	40	31	0	0	0	0	0
28	31	40	41	32	0	0	0	0	0
29	32	41	42	33	0	0	0	0	0

I	30	I	33	42	43	34	0	0	0	0	0 I
I	31	I	34	43	44	35	0	0	0	0	0 I
I	32	I	35	44	45	36	0	0	0	0	0 I

STRUCTURE EN LIGNE DE CIEL

EQ.#	LCIEL1(I)	DIAG1(I)
1	0	1
2	1	4
3	2	9
4	3	16
5	4	25
6	5	36
7	3	43
8	4	52
9	5	63
10	3	70
11	4	79
12	5	90
13	3	97
14	4	106
15	5	117
16	3	124
17	4	133
18	5	144
19	3	151
20	4	160
21	5	171
22	3	178
23	4	187
24	5	198
25	3	205
26	4	214
27	5	225
28	27	280
29	28	337
30	29	396
31	30	457
32	31	520
33	32	585
34	30	646
35	31	709
36	32	774
37	30	835
38	31	898
39	32	963
40	30	1024
41	31	1087
42	32	1152
43	30	1213
44	31	1276
45	32	1341
46	30	1402
47	31	1465
48	32	1530
49	30	1591
50	31	1654
51	32	1719
52	30	1780
53	31	1843
54	32	1908
55	27	1963

56	28	2020
57	29	2079
58	30	2140
59	31	2203
60	32	2268
61	30	2329
62	31	2392
63	32	2457
64	30	2518
65	31	2581
66	32	2646
67	30	2707
68	31	2770
69	32	2835
70	30	2896
71	31	2959
72	32	3024
73	30	3085
74	31	3148
75	32	3213
76	30	3274
77	31	3337
78	32	3402
79	30	3463
80	31	3526
81	32	3591
82	27	3646
83	28	3703
84	29	3762
85	30	3823
86	31	3886
87	32	3951
88	30	4012
89	31	4075
90	32	4140
91	30	4201
92	31	4264
93	32	4329
94	30	4390
95	31	4453
96	32	4518
97	30	4579
98	31	4642
99	32	4707
100	30	4768
101	31	4831
102	32	4896
103	30	4957
104	31	5020
105	32	5085
106	30	5146
107	31	5209
108	32	5274

*** LISTBL *** APPELE DE LA ROUTINE : " INDATA"

LISTE DE TOUS LES TABLEAUX ALLOUES A CET INSTANT

NO.	TABLE	LONGUEUR	TYPE	DEBUT A	FIN A	ORIGINE
1 -	COORD	306	REAL	1	612	INDATA
2 -	COORD1	90	REAL	613	792	INDATA
3 -	IDEQ	306	INTEGER	793	1098	INDATA

4 -	IDEQ1	135	INTEGER	1099	1234	INDATA
5 -	CONSTR	16	REAL	1235	1266	INDATA
6 -	CONSTR1	26	REAL	1267	1318	INDATA
7 -	CONNEC	288	INTEGER	1319	1606	INDATA
8 -	CONNEC1	128	INTEGER	1607	1734	INDATA
9 -	LMGLOB	576	INTEGER	1735	2310	INDATA
10 -	LMGLOB1	384	INTEGER	2311	2694	INDATA
11 -	DIAG	233	INTEGER	2695	2928	INDATA
12 -	DIAG1	108	INTEGER	2929	3036	INDATA

*** LISTBL *** APPELE DE LA ROUTINE : " MODSOL"

LISTE DE TOUS LES TABLEAUX ALLOUES A CET INSTANT

NO.	TABLE	LONGUEUR	TYPE	DEBUT A	FIN A	ORIGINE
1 -	COORD	306	REAL	1	612	INDATA
2 -	COORD1	90	REAL	613	792	INDATA
3 -	IDEQ	306	INTEGER	793	1098	INDATA
4 -	IDEQ1	135	INTEGER	1099	1234	INDATA
5 -	CONSTR	16	REAL	1235	1266	INDATA
6 -	CONSTR1	26	REAL	1267	1318	INDATA
7 -	CONNEC	288	INTEGER	1319	1606	INDATA
8 -	CONNEC1	128	INTEGER	1607	1734	INDATA
9 -	LMGLOB	576	INTEGER	1735	2310	INDATA
10 -	LMGLOB1	384	INTEGER	2311	2694	INDATA
11 -	DIAG	233	INTEGER	2695	2928	INDATA
12 -	DIAG1	108	INTEGER	2929	3036	INDATA
13 -	A	18853	REAL	3037	40742	MODSOL
14 -	A1	5274	REAL	40743	51290	MODSOL
15 -	X	233	REAL	51291	51756	MODSOL
16 -	X1	108	REAL	51757	51972	MODSOL
17 -	B	233	REAL	51973	52438	MODSOL
18 -	B1	108	REAL	52439	52654	MODSOL
19 -	DX	233	REAL	52655	53120	MODSOL
20 -	DX1	108	REAL	53121	53336	MODSOL

EQUATIONS DU MOUVEMENT

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE			VALEUR MAXIMALE		
	X	DX	R	X	DX	R
1	8.508	0.8981E-01	0.20E-01	1.000	0.1507E-01	0.39E-02
2	8.492	0.2972E-01	0.68E-02	1.000	0.4994E-02	0.13E-02
3	8.497	0.9832E-02	0.22E-02	1.000	0.1655E-02	0.44E-03
4	8.495	0.3253E-02	0.76E-03	1.000	0.5487E-03	0.15E-03
5	8.496	0.1077E-02	0.25E-03	1.000	0.1819E-03	0.49E-04
6	8.496	0.3563E-03	0.84E-04	1.000	0.6030E-04	0.16E-04
7	8.496	0.1179E-03	0.27E-04	1.000	0.1998E-04	0.54E-05
8	8.496	0.3900E-04	0.94E-05	1.000	0.6621E-05	0.18E-05
9	8.496	0.1291E-04	0.30E-05	1.000	0.2196E-05	0.60E-06
10	8.496	0.4283E-05	0.10E-05	1.000	0.7311E-06	0.20E-06

EQUATIONS DE CONFORMATION

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE			VALEUR MAXIMALE		
	X	DX	R	X	DX	R
1	2.839	I0.5387	0.24E-01	I0.3411	I0.1602	0.82E-02
2	2.638	I0.8192E-03	0.32E-04	I0.3536	I0.3797E-03	0.11E-04
3	2.638	I0.3928E-05	0.14E-06	I0.3535	I0.1864E-05	0.46E-07
4	2.638	I0.3359E-07	0.14E-08	I0.3535	I0.1480E-07	0.51E-09

EQUATIONS DU MOUVEMENT

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE			VALEUR MAXIMALE		
	X	DX	R	X	DX	R
1	8.496	I0.9964E-01	0.11	I 1.000	I0.1762E-01	0.34E-01
2	8.482	I0.3299E-01	0.37E-01	I0.9993	I0.5846E-02	0.11E-01
3	8.486	I0.1092E-01	0.12E-01	I0.9995	I0.1939E-02	0.38E-02
4	8.485	I0.3618E-02	0.41E-02	I0.9994	I0.6433E-03	0.13E-02
5	8.485	I0.1198E-02	0.14E-02	I0.9995	I0.2134E-03	0.42E-03
6	8.485	I0.3970E-03	0.45E-03	I0.9995	I0.7080E-04	0.14E-03
7	8.485	I0.1315E-03	0.15E-03	I0.9995	I0.2349E-04	0.46E-04
8	8.485	I0.4358E-04	0.50E-04	I0.9995	I0.7793E-05	0.16E-04
9	8.485	I0.1443E-04	0.17E-04	I0.9995	I0.2586E-05	0.52E-05
10	8.485	I0.4790E-05	0.56E-05	I0.9995	I0.8580E-06	0.17E-05
11	8.485	I0.1586E-05	0.19E-05	I0.9995	I0.2849E-06	0.57E-06

EQUATIONS DE CONFORMATION

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE			VALEUR MAXIMALE		
	X	DX	R	X	DX	R
1	2.638	I0.3875E-02	0.30E-03	I0.3535	I0.1167E-02	0.61E-04
2	2.639	I0.1572E-03	0.10E-04	I0.3528	I0.6701E-04	0.38E-05
3	2.639	I0.2197E-05	0.11E-06	I0.3527	I0.1056E-05	0.53E-07
4	2.639	I0.2972E-07	0.14E-08	I0.3527	I0.1402E-07	0.72E-09

ITERATION VISCOELASTIQUE # 2 ERREUR = 0.92E-01

EQUATIONS DU MOUVEMENT

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE			VALEUR MAXIMALE		
	X	DX	R	X	DX	R
1	8.485	I0.2783E-01	0.99E-03	I0.9995	I0.4299E-02	0.32E-03
2	8.480	I0.9202E-02	0.31E-03	I0.9993	I0.1423E-02	0.96E-04
3	8.482	I0.3043E-02	0.11E-03	I0.9994	I0.4712E-03	0.35E-04
4	8.481	I0.1006E-02	0.34E-04	I0.9994	I0.1560E-03	0.11E-04

5	8.481	0.3327E-03	0.12E-04	0.9994	0.5166E-04	0.38E-05
6	8.481	0.1100E-03	0.37E-05	0.9994	0.1710E-04	0.12E-05
7	8.481	0.3638E-04	0.13E-05	0.9994	0.5661E-05	0.41E-06
8	8.481	0.1203E-04	0.41E-06	0.9994	0.1874E-05	0.13E-06
9	8.481	0.3989E-05	0.14E-06	0.9994	0.6214E-06	0.45E-07

EQUATIONS DE CONFORMATION

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE			VALEUR MAXIMALE		
	X	DX	R	X	DX	R
1	2.639	0.1064E-02	0.88E-04	0.3527	0.3342E-03	0.20E-04
2	2.639	0.4296E-04	0.28E-05	0.3528	0.1930E-04	0.95E-06
3	2.639	0.5766E-06	0.29E-07	0.3528	0.2901E-06	0.14E-07

ITERATION VISCOELASTIQUE # 3 ERREUR = 0.32E-02

EQUATIONS DU MOUVEMENT

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE			VALEUR MAXIMALE		
	X	DX	R	X	DX	R
1	8.481	0.1032E-01	0.28E-03	0.9994	0.1612E-02	0.10E-03
2	8.479	0.3415E-02	0.85E-04	0.9993	0.5339E-03	0.28E-04
3	8.480	0.1129E-02	0.31E-04	0.9993	0.1768E-03	0.11E-04
4	8.480	0.3736E-03	0.92E-05	0.9993	0.5857E-04	0.30E-05
5	8.480	0.1236E-03	0.34E-05	0.9993	0.1940E-04	0.11E-05
6	8.480	0.4090E-04	0.10E-05	0.9993	0.6432E-05	0.32E-06
7	8.480	0.1353E-04	0.37E-06	0.9993	0.2129E-05	0.12E-06
8	8.480	0.4466E-05	0.11E-06	0.9993	0.6988E-06	0.34E-07

EQUATIONS DE CONFORMATION

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE			VALEUR MAXIMALE		
	X	DX	R	X	DX	R
1	2.639	0.4441E-03	0.33E-04	0.3528	0.1492E-03	0.72E-05
2	2.639	0.1411E-04	0.93E-06	0.3529	0.6713E-05	0.31E-06
3	2.639	0.1837E-06	0.93E-08	0.3529	0.1016E-06	0.46E-08

ITERATION VISCOELASTIQUE # 4 ERREUR = 0.10E-02

EQUATIONS DU MOUVEMENT

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE	VALEUR MAXIMALE

	X	DX	R	X	DX	R
1	8.480	I0.3904E-02	0.11E-03 I0.9993	I0.6336E-03	0.40E-04	
2	8.479	I0.1292E-02	0.32E-04 I0.9993	I0.2100E-03	0.11E-04	
3	8.479	I0.4277E-03	0.12E-04 I0.9993	I0.6961E-04	0.44E-05	
4	8.479	I0.1415E-03	0.35E-05 I0.9993	I0.2307E-04	0.12E-05	
5	8.479	I0.4686E-04	0.14E-05 I0.9993	I0.7650E-05	0.48E-06	
6	8.479	I0.1551E-04	0.38E-06 I0.9993	I0.2538E-05	0.13E-06	
7	8.479	I0.5129E-05	0.15E-06 I0.9993	I0.8425E-06	0.53E-07	

EQUATIONS DE CONFORMATION

CALCUL DES NORMES

ITER	NORME EUCLEDIENNE			VALEUR MAXIMALE		
	X	DX	R	X	DX	R
1	2.639	I0.1788E-03	0.12E-04 I0.3529	I0.7077E-04	0.27E-05	
2	2.639	I0.4974E-05	0.32E-06 I0.3529	I0.2479E-05	0.11E-06	
3	2.639	I0.6333E-07	0.31E-08 I0.3529	I0.3618E-07	0.16E-08	

5 ITERATIONS VISCOELASTIQUES: *** CONVERGENCE ***

NORME MAXIMALE X(I)-X(I-1) : 0.01529
NORME EUCLEDIENNE X(I)-X(I-1) : 0.09782
NORME MAXIMALE X1(I)-X1(I-1) : 0.15973
NORME EUCLEDIENNE X1(I)-X1(I-1) : 0.53720

} écarts estimés initiaux et solution calculée

SOLUTION DES VARIABLES PRIMAIRES

NOEUD #	VARIABLES DEPENDANTES U	V	W	T	COORDONNEES COORD #1	COORD #2	COORD #3
1	1.00000	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
2	0.99612	0.00000	0.00000	0.00000	0.00000	0.06250	0.00000
3	0.98447	0.00000	0.00000	0.00000	0.00000	0.12500	0.00000
4	0.96504	0.00000	0.00000	0.00000	0.00000	0.18750	0.00000
5	0.93785	0.00000	0.00000	0.00000	0.00000	0.25000	0.00000
6	0.90286	0.00000	0.00000	0.00000	0.00000	0.31250	0.00000
7	0.86009	0.00000	0.00000	0.00000	0.00000	0.37500	0.00000
8	0.80951	0.00000	0.00000	0.00000	0.00000	0.43750	0.00000
9	0.75110	0.00000	0.00000	0.00000	0.00000	0.50000	0.00000
10	0.68487	0.00000	0.00000	0.00000	0.00000	0.56250	0.00000
11	0.61078	0.00000	0.00000	0.00000	0.00000	0.62500	0.00000
12	0.52881	0.00000	0.00000	0.00000	0.00000	0.68750	0.00000
13	0.43895	0.00000	0.00000	0.00000	0.00000	0.75000	0.00000
14	0.34117	0.00000	0.00000	0.00000	0.00000	0.81250	0.00000
15	0.23543	0.00000	0.00000	0.00000	0.00000	0.87500	0.00000
16	0.12172	0.00000	0.00000	0.00000	0.00000	0.93750	0.00000
17	0.00000	0.00000	0.00000	0.00000	0.00000	1.00000	0.00000
18	0.99930	0.00000	0.00000	0.00000	0.25000	0.00000	0.00000
19	0.99544	0.00038	0.00000	0.00000	0.25000	0.06250	0.00000
20	0.98387	0.00073	0.00000	0.00000	0.25000	0.12500	0.00000
21	0.96456	0.00103	0.00000	0.00000	0.25000	0.18750	0.00000
22	0.93750	0.00127	0.00000	0.00000	0.25000	0.25000	0.00000
23	0.90269	0.00142	0.00000	0.00000	0.25000	0.31250	0.00000
24	0.86008	0.00148	0.00000	0.00000	0.25000	0.37500	0.00000
25	0.80966	0.00145	0.00000	0.00000	0.25000	0.43750	0.00000

I	26 I	0.75140	0.00133	0.00000	0.00000 I	0.25000	0.50000	0.00000 I
I	27 I	0.68527	0.00115	0.00000	0.00000 I	0.25000	0.56250	0.00000 I
I	28 I	0.61125	0.00092	0.00000	0.00000 I	0.25000	0.62500	0.00000 I
I	29 I	0.52931	0.00066	0.00000	0.00000 I	0.25000	0.68750	0.00000 I
I	30 I	0.43938	0.00041	0.00000	0.00000 I	0.25000	0.75000	0.00000 I
I	31 I	0.34147	0.00020	0.00000	0.00000 I	0.25000	0.81250	0.00000 I
I	32 I	0.23554	0.00006	0.00000	0.00000 I	0.25000	0.87500	0.00000 I
I	33 I	0.12170	0.00001	0.00000	0.00000 I	0.25000	0.93750	0.00000 I
I	34 I	0.00000	0.00000	0.00000	0.00000 I	0.25000	1.00000	0.00000 I
I	35 I	0.99667	0.00000	0.00000	0.00000 I	0.50000	0.00000	0.00000 I
I	36 I	0.99289	0.00096	0.00000	0.00000 I	0.50000	0.06250	0.00000 I
I	37 I	0.98157	0.00186	0.00000	0.00000 I	0.50000	0.12500	0.00000 I
I	38 I	0.96265	0.00264	0.00000	0.00000 I	0.50000	0.18750	0.00000 I
I	39 I	0.93611	0.00326	0.00000	0.00000 I	0.50000	0.25000	0.00000 I
I	40 I	0.90189	0.00368	0.00000	0.00000 I	0.50000	0.31250	0.00000 I
I	41 I	0.85991	0.00387	0.00000	0.00000 I	0.50000	0.37500	0.00000 I
I	42 I	0.81010	0.00384	0.00000	0.00000 I	0.50000	0.43750	0.00000 I
I	43 I	0.75237	0.00359	0.00000	0.00000 I	0.50000	0.50000	0.00000 I
I	44 I	0.68664	0.00317	0.00000	0.00000 I	0.50000	0.56250	0.00000 I
I	45 I	0.61286	0.00264	0.00000	0.00000 I	0.50000	0.62500	0.00000 I
I	46 I	0.53098	0.00206	0.00000	0.00000 I	0.50000	0.68750	0.00000 I
I	47 I	0.44095	0.00148	0.00000	0.00000 I	0.50000	0.75000	0.00000 I
I	48 I	0.34279	0.00094	0.00000	0.00000 I	0.50000	0.81250	0.00000 I
I	49 I	0.23653	0.00047	0.00000	0.00000 I	0.50000	0.87500	0.00000 I
I	50 I	0.12216	0.00012	0.00000	0.00000 I	0.50000	0.93750	0.00000 I
I	51 I	0.00000	0.00000	0.00000	0.00000 I	0.50000	1.00000	0.00000 I
I	52 I	0.99132	0.00000	0.00000	0.00000 I	0.75000	0.00000	0.00000 I
I	53 I	0.98768	0.00152	0.00000	0.00000 I	0.75000	0.06250	0.00000 I
I	54 I	0.97674	0.00297	0.00000	0.00000 I	0.75000	0.12500	0.00000 I
I	55 I	0.95846	0.00431	0.00000	0.00000 I	0.75000	0.18750	0.00000 I
I	56 I	0.93278	0.00546	0.00000	0.00000 I	0.75000	0.25000	0.00000 I
I	57 I	0.89961	0.00636	0.00000	0.00000 I	0.75000	0.31250	0.00000 I
I	58 I	0.85882	0.00698	0.00000	0.00000 I	0.75000	0.37500	0.00000 I
I	59 I	0.81026	0.00726	0.00000	0.00000 I	0.75000	0.43750	0.00000 I
I	60 I	0.75377	0.00717	0.00000	0.00000 I	0.75000	0.50000	0.00000 I
I	61 I	0.68916	0.00670	0.00000	0.00000 I	0.75000	0.56250	0.00000 I
I	62 I	0.61625	0.00589	0.00000	0.00000 I	0.75000	0.62500	0.00000 I
I	63 I	0.53486	0.00478	0.00000	0.00000 I	0.75000	0.68750	0.00000 I
I	64 I	0.44493	0.00350	0.00000	0.00000 I	0.75000	0.75000	0.00000 I
I	65 I	0.34640	0.00220	0.00000	0.00000 I	0.75000	0.81250	0.00000 I
I	66 I	0.23929	0.00108	0.00000	0.00000 I	0.75000	0.87500	0.00000 I
I	67 I	0.12387	0.00030	0.00000	0.00000 I	0.75000	0.93750	0.00000 I
I	68 I	0.00000	0.00000	0.00000	0.00000 I	0.75000	1.00000	0.00000 I
I	69 I	0.98586	0.00000	0.00000	0.00000 I	1.00000	0.00000	0.00000 I
I	70 I	0.98229	0.00103	0.00000	0.00000 I	1.00000	0.06250	0.00000 I
I	71 I	0.97159	0.00204	0.00000	0.00000 I	1.00000	0.12500	0.00000 I
I	72 I	0.95372	0.00305	0.00000	0.00000 I	1.00000	0.18750	0.00000 I
I	73 I	0.92862	0.00401	0.00000	0.00000 I	1.00000	0.25000	0.00000 I
I	74 I	0.89622	0.00492	0.00000	0.00000 I	1.00000	0.31250	0.00000 I
I	75 I	0.85640	0.00573	0.00000	0.00000 I	1.00000	0.37500	0.00000 I
I	76 I	0.80902	0.00638	0.00000	0.00000 I	1.00000	0.43750	0.00000 I
I	77 I	0.75391	0.00682	0.00000	0.00000 I	1.00000	0.50000	0.00000 I
I	78 I	0.69083	0.00694	0.00000	0.00000 I	1.00000	0.56250	0.00000 I
I	79 I	0.61949	0.00667	0.00000	0.00000 I	1.00000	0.62500	0.00000 I
I	80 I	0.53958	0.00592	0.00000	0.00000 I	1.00000	0.68750	0.00000 I
I	81 I	0.45074	0.00467	0.00000	0.00000 I	1.00000	0.75000	0.00000 I
I	82 I	0.35237	0.00305	0.00000	0.00000 I	1.00000	0.81250	0.00000 I
I	83 I	0.24426	0.00141	0.00000	0.00000 I	1.00000	0.87500	0.00000 I
I	84 I	0.12643	0.00032	0.00000	0.00000 I	1.00000	0.93750	0.00000 I
I	85 I	0.00000	0.00000	0.00000	0.00000 I	1.00000	1.00000	0.00000 I
I	86 I	0.98471	0.00000	0.00000	0.00000 I	1.12500	0.00000	0.00000 I
I	87 I	0.98114	0.00010	0.00000	0.00000 I	1.12500	0.06250	0.00000 I
I	88 I	0.97041	0.00022	0.00000	0.00000 I	1.12500	0.12500	0.00000 I
I	89 I	0.95249	0.00038	0.00000	0.00000 I	1.12500	0.18750	0.00000 I
I	90 I	0.92735	0.00061	0.00000	0.00000 I	1.12500	0.25000	0.00000 I
I	91 I	0.89492	0.00092	0.00000	0.00000 I	1.12500	0.31250	0.00000 I

92	0.85511	0.00130	0.00000	0.00000	1.12500	0.37500	0.00000
93	0.80782	0.00177	0.00000	0.00000	1.12500	0.43750	0.00000
94	0.75291	0.00229	0.00000	0.00000	1.12500	0.50000	0.00000
95	0.69019	0.00283	0.00000	0.00000	1.12500	0.56250	0.00000
96	0.61939	0.00331	0.00000	0.00000	1.12500	0.62500	0.00000
97	0.54025	0.00365	0.00000	0.00000	1.12500	0.68750	0.00000
98	0.45233	0.00371	0.00000	0.00000	1.12500	0.75000	0.00000
99	0.35501	0.00327	0.00000	0.00000	1.12500	0.81250	0.00000
100	0.24731	0.00229	0.00000	0.00000	1.12500	0.87500	0.00000
101	0.12902	0.00085	0.00000	0.00000	1.12500	0.93750	0.00000
102	0.00000	0.00000	0.00000	0.00000	1.12500	1.00000	0.00000
103	0.98558	0.00000	0.00000	0.00000	1.25000	0.00000	0.00000
104	0.98198	-0.00098	0.00000	0.00000	1.25000	0.06250	0.00000
105	0.97118	-0.00193	0.00000	0.00000	1.25000	0.12500	0.00000
106	0.95313	-0.00284	0.00000	0.00000	1.25000	0.18750	0.00000
107	0.92783	-0.00369	0.00000	0.00000	1.25000	0.25000	0.00000
108	0.89518	-0.00445	0.00000	0.00000	1.25000	0.31250	0.00000
109	0.85514	-0.00509	0.00000	0.00000	1.25000	0.37500	0.00000
110	0.80758	-0.00557	0.00000	0.00000	1.25000	0.43750	0.00000
111	0.75240	-0.00587	0.00000	0.00000	1.25000	0.50000	0.00000
112	0.68941	-0.00593	0.00000	0.00000	1.25000	0.56250	0.00000
113	0.61841	-0.00570	0.00000	0.00000	1.25000	0.62500	0.00000
114	0.53915	-0.00514	0.00000	0.00000	1.25000	0.68750	0.00000
115	0.45127	-0.00418	0.00000	0.00000	1.25000	0.75000	0.00000
116	0.35450	-0.00280	0.00000	0.00000	1.25000	0.81250	0.00000
117	0.24773	-0.00147	0.00000	0.00000	1.25000	0.87500	0.00000
118	0.13011	-0.00047	0.00000	0.00000	1.25000	0.93750	0.00000
119	0.00000	0.00000	0.00000	0.00000	1.25000	1.00000	0.00000
120	0.98666	0.00000	0.00000	0.00000	1.31250	0.00000	0.00000
121	0.98305	-0.00101	0.00000	0.00000	1.31250	0.06250	0.00000
122	0.97223	-0.00201	0.00000	0.00000	1.31250	0.12500	0.00000
123	0.95414	-0.00299	0.00000	0.00000	1.31250	0.18750	0.00000
124	0.92878	-0.00394	0.00000	0.00000	1.31250	0.25000	0.00000
125	0.89606	-0.00484	0.00000	0.00000	1.31250	0.31250	0.00000
126	0.85591	-0.00568	0.00000	0.00000	1.31250	0.37500	0.00000
127	0.80821	-0.00642	0.00000	0.00000	1.31250	0.43750	0.00000
128	0.75284	-0.00704	0.00000	0.00000	1.31250	0.50000	0.00000
129	0.68959	-0.00747	0.00000	0.00000	1.31250	0.56250	0.00000
130	0.61828	-0.00765	0.00000	0.00000	1.31250	0.62500	0.00000
131	0.53858	-0.00747	0.00000	0.00000	1.31250	0.68750	0.00000
132	0.45021	-0.00686	0.00000	0.00000	1.31250	0.75000	0.00000
133	0.35270	-0.00552	0.00000	0.00000	1.31250	0.81250	0.00000
134	0.24570	-0.00341	0.00000	0.00000	1.31250	0.87500	0.00000
135	0.12845	-0.00111	0.00000	0.00000	1.31250	0.93750	0.00000
136	0.00000	0.00000	0.00000	0.00000	1.31250	1.00000	0.00000
137	0.98726	0.00000	0.00000	0.00000	1.37500	0.00000	0.00000
138	0.98364	0.00000	0.00000	0.00000	1.37500	0.06250	0.00000
139	0.97281	0.00000	0.00000	0.00000	1.37500	0.12500	0.00000
140	0.95471	0.00000	0.00000	0.00000	1.37500	0.18750	0.00000
141	0.92934	0.00000	0.00000	0.00000	1.37500	0.25000	0.00000
142	0.89658	0.00000	0.00000	0.00000	1.37500	0.31250	0.00000
143	0.85640	0.00000	0.00000	0.00000	1.37500	0.37500	0.00000
144	0.80863	0.00000	0.00000	0.00000	1.37500	0.43750	0.00000
145	0.75319	0.00000	0.00000	0.00000	1.37500	0.50000	0.00000
146	0.68980	0.00000	0.00000	0.00000	1.37500	0.56250	0.00000
147	0.61834	0.00000	0.00000	0.00000	1.37500	0.62500	0.00000
148	0.53838	0.00000	0.00000	0.00000	1.37500	0.68750	0.00000
149	0.44976	0.00000	0.00000	0.00000	1.37500	0.75000	0.00000
150	0.35163	0.00000	0.00000	0.00000	1.37500	0.81250	0.00000
151	0.24420	0.00000	0.00000	0.00000	1.37500	0.87500	0.00000
152	0.12721	0.00000	0.00000	0.00000	1.37500	0.93750	0.00000
153	0.00000	0.00000	0.00000	0.00000	1.37500	1.00000	0.00000

SOLUTION DES VARIABLES PRIMAIRES

NOEUD #	VARIABLES DEPENDANTES CXX	CYY	CXY	COORDONNEES COORD #1	COORD #2	COORD #3
1	0.33333	0.33333	0.00000	0.00000	0.00000	0.00000
2	0.33346	0.33327	-0.00555	0.00000	0.12500	0.00000
3	0.33383	0.33309	-0.01110	0.00000	0.25000	0.00000
4	0.33444	0.33278	-0.01664	0.00000	0.37500	0.00000
5	0.33530	0.33235	-0.02216	0.00000	0.50000	0.00000
6	0.33641	0.33180	-0.02765	0.00000	0.62500	0.00000
7	0.33775	0.33113	-0.03311	0.00000	0.75000	0.00000
8	0.33933	0.33034	-0.03854	0.00000	0.87500	0.00000
9	0.34114	0.32943	-0.04392	0.00000	1.00000	0.00000
10	0.33239	0.33358	0.00000	0.50000	0.00000	0.00000
11	0.33260	0.33346	-0.00530	0.50000	0.12500	0.00000
12	0.33320	0.33311	-0.01065	0.50000	0.25000	0.00000
13	0.33420	0.33262	-0.01606	0.50000	0.37500	0.00000
14	0.33555	0.33214	-0.02159	0.50000	0.50000	0.00000
15	0.33718	0.33185	-0.02725	0.50000	0.62500	0.00000
16	0.33907	0.33202	-0.03305	0.50000	0.75000	0.00000
17	0.34090	0.33205	-0.03874	0.50000	0.87500	0.00000
18	0.34204	0.33036	-0.04411	0.50000	1.00000	0.00000
19	0.33346	0.33514	0.00000	1.00000	0.00000	0.00000
20	0.33357	0.33500	-0.00535	1.00000	0.12500	0.00000
21	0.33388	0.33455	-0.01074	1.00000	0.25000	0.00000
22	0.33437	0.33371	-0.01618	1.00000	0.37500	0.00000
23	0.33501	0.33235	-0.02169	1.00000	0.50000	0.00000
24	0.33573	0.33037	-0.02727	1.00000	0.62500	0.00000
25	0.33650	0.32791	-0.03281	1.00000	0.75000	0.00000
26	0.33741	0.32590	-0.03865	1.00000	0.87500	0.00000
27	0.34069	0.32901	-0.04413	1.00000	1.00000	0.00000
28	0.32339	0.32309	0.00000	1.25000	0.00000	0.00000
29	0.32417	0.32380	-0.00444	1.25000	0.12500	0.00000
30	0.32650	0.32590	-0.00907	1.25000	0.25000	0.00000
31	0.33033	0.32930	-0.01414	1.25000	0.37500	0.00000
32	0.33557	0.33376	-0.01984	1.25000	0.50000	0.00000
33	0.34191	0.33870	-0.02644	1.25000	0.62500	0.00000
34	0.34829	0.34286	-0.03386	1.25000	0.75000	0.00000
35	0.35292	0.34304	-0.04221	1.25000	0.87500	0.00000
36	0.34354	0.33001	-0.04643	1.25000	1.00000	0.00000
37	0.17383	0.17360	0.00000	1.37500	0.00000	0.00000
38	0.17519	0.17495	-0.00128	1.37500	0.12500	0.00000
39	0.17929	0.17900	-0.00270	1.37500	0.25000	0.00000
40	0.18642	0.18602	-0.00444	1.37500	0.37500	0.00000
41	0.19747	0.19686	-0.00679	1.37500	0.50000	0.00000
42	0.21347	0.21252	-0.01017	1.37500	0.62500	0.00000
43	0.23833	0.23645	-0.01584	1.37500	0.75000	0.00000
44	0.27203	0.26892	-0.02503	1.37500	0.87500	0.00000
45	0.34596	0.33316	-0.04674	1.37500	1.00000	0.00000

SOLUTION DES VARIABLES SECONDAIRES

IELEMENT #	VARIABLES DEPENDANTES PRESSION				COORDONNEES COORD #1	COORD #2	COORD #3
1	0.169E+00	0.130E+00	0.169E+00	0.130E+00	0.25000	0.06250	0.00000
2	0.101E+00	0.641E-01	0.101E+00	0.638E-01	0.75000	0.06250	0.00000
3	0.169E+00	0.130E+00	0.169E+00	0.130E+00	0.25000	0.18750	0.00000
4	0.101E+00	0.634E-01	0.101E+00	0.627E-01	0.75000	0.18750	0.00000
5	0.169E+00	0.130E+00	0.169E+00	0.130E+00	0.25000	0.31250	0.00000

6	0.101E+00	0.620E-01	0.101E+00	0.610E-01	0.75000	0.31250	0.00000
7	0.169E+00	0.130E+00	0.168E+00	0.129E+00	0.25000	0.43750	0.00000
8	0.100E+00	0.602E-01	0.100E+00	0.592E-01	0.75000	0.43750	0.00000
9	0.168E+00	0.129E+00	0.168E+00	0.128E+00	0.25000	0.56250	0.00000
10	0.100E+00	0.586E-01	0.998E-01	0.582E-01	0.75000	0.56250	0.00000
11	0.168E+00	0.128E+00	0.167E+00	0.126E+00	0.25000	0.68750	0.00000
12	0.995E-01	0.581E-01	0.989E-01	0.587E-01	0.75000	0.68750	0.00000
13	0.167E+00	0.126E+00	0.167E+00	0.125E+00	0.25000	0.81250	0.00000
14	0.985E-01	0.594E-01	0.980E-01	0.608E-01	0.75000	0.81250	0.00000
15	0.167E+00	0.126E+00	0.168E+00	0.126E+00	0.25000	0.93750	0.00000
16	0.976E-01	0.605E-01	0.986E-01	0.561E-01	0.75000	0.93750	0.00000
17	0.526E-01	0.538E-01	0.519E-01	0.525E-01	1.12500	0.06250	0.00000
18	0.509E-01	0.505E-01	0.489E-01	0.464E-01	1.12500	0.18750	0.00000
19	0.470E-01	0.424E-01	0.438E-01	0.356E-01	1.12500	0.31250	0.00000
20	0.412E-01	0.298E-01	0.371E-01	0.204E-01	1.12500	0.43750	0.00000
21	0.340E-01	0.128E-01	0.299E-01	0.140E-02	1.12500	0.56250	0.00000
22	0.271E-01	-0.717E-02	0.243E-01	-0.190E-01	1.12500	0.68750	0.00000
23	0.231E-01	-0.265E-01	0.244E-01	-0.342E-01	1.12500	0.81250	0.00000
24	0.261E-01	-0.326E-01	0.311E-01	-0.183E-01	1.12500	0.93750	0.00000
25	0.145E+00	0.390E+00	0.143E+00	0.388E+00	1.31250	0.06250	0.00000
26	0.141E+00	0.384E+00	0.136E+00	0.378E+00	1.31250	0.18750	0.00000
27	0.130E+00	0.371E+00	0.122E+00	0.360E+00	1.31250	0.31250	0.00000
28	0.114E+00	0.349E+00	0.101E+00	0.332E+00	1.31250	0.43750	0.00000
29	0.905E-01	0.316E+00	0.734E-01	0.291E+00	1.31250	0.56250	0.00000
30	0.593E-01	0.268E+00	0.373E-01	0.229E+00	1.31250	0.68750	0.00000
31	0.208E-01	0.197E+00	-0.496E-02	0.144E+00	1.31250	0.81250	0.00000
32	-0.243E-01	0.841E-01	-0.426E-01	-0.186E-01	1.31250	0.93750	0.00000

```
* ce programme calcule la solution analytique
* pour l'ecoulement du fluide de grmela entre 2 plaques
* il construit le fichier des c.f. adimensionnelles  pour le programme
* nn2dgr (stoke dans le fichier CF)
*
*
      implicit none
      integer i, j, nbrnde, nbrnde1, flag, flag1
      integer no(100), no1(50), no2(100)
      integer compteur, nbrnde2
      real*8 sxy, umax ,umoy, h1, h2, pas, denom
      real*8 gradp, etha0, lambda, debit, de, tauxcis
      real*8 y(100), y1(50), u$(100), v$(100), u2$(100)
      real*8 cxx$(50), cyy$(50), cxy$(50)
      real*8 a, b, gradp2, erreur, precedent
      character*75 ligne
      data compteur / 0 /
      open(file='CF', unit=3)
      open(file='vitesses.no',unit=1)
      open(file='vitesses.y', unit=2)
      open(file='conform.no', unit=7)
      open(file='conform.y',  unit=8)
      open(file='vitesses.no2',unit=9)
      write(*,*) ' nombre de noeuds de vitesses section d"entree ?'
      read(*,*) nbrnde
      nbrnde1 = nbrnde/2 + 1
      write(*,*) ' nombre de noeuds de conformation =',nbrnde1
      write(*,*) ' gradient de pression (negatif) ? '
      read(*,*) gradp
      write(*,*) ' etha0 ?'
      read(*,*) etha0
      write(*,*) ' temps caracteristique du fluide ?'
      read(*,*) lambda
      write(*,*) ' demie hauteur du canal amont ?'
      read(*,*) h1
      write(*,*) ' demie hauteur du canal aval ?'
      read(*,*) h2
      write(*,*) ' espacement regulier ?, oui=1, non=0'
      read(*,*) flag
      if(flag.eq.1) then
        write(*,*) 'entrez le pas (vitesse)'
        read(*,*) pas
        nbrnde2=1
        y(1)=0.0d0
        y1(1)=0.0d0
        no(1)=1
        no1(1)=1
        no2(1)=51
        do 5 i=2,nbrnde
          y(i)=y(i-1)+pas
          no(i)=no(i-1)+51
          if(y(i).lt.h2) then
            no2(i)=no2(i-1)+51
            nbrnde2=nbrnde2 + 1
          endif
5       continue
        do 6 i=2,nbrnde1
          y1(i)=y1(i-1) + 2.0d0*pas
          no1(i)=no1(i-1)+26
6       continue
        if(dabs(y(nbrnde)-h1).gt.1.0d-8) then
          write(*,1003) y(nbrnde),h1
          stop
        endif
      elseif(flag.eq.0) then
        write(*,5000)
```

```
        read(*,*) flag1
        if(flag1.eq.0) then
          nbrnde2=0
          write(*,*) ' entrer les # des noeuds de vitesses (entree)'
          read(*,*)  (no(i), i=1,nbrnde)
*          write(1,*) ' numeros des noeuds de vitesses a l"entree'
*          write(1,*) (no(i), i=1,nbrnde)
          write(*,*) ' entrer les ordonnees des noeuds de vitesses'
          read(*,*)  (y(i),  i=1,nbrnde)
          do 80 i=1,nbrnde
            if(y(i).lt.h2) nbrnde2=nbrnde2+1
 80       continue
*          write(2,*) ' ordonnees des noeuds de vitesses a l"entree'
*          write(2,*)  (y(i),  i=1,nbrnde)
          write(*,*) ' entrer les # des noeuds de vitesses (sortie)'
          read(*,*) (no2(i), i=1,nbrnde2)
*          write(9,*) ' numeros des noeuds de vitesses a la sortie'
*          write(9,*) (no2(i), i=1,nbrnde2)
          write(*,*) ' entrer les # des noeuds de conformation (entree)'
          read(*,*) (no1(i),  i=1,nbrnde1)
*          write(7,*) ' numeros des noeuds de conformation a l"entree'
*          write(7,*) (no1(i),  i=1,nbrnde1)
          write(*,*) ' entrer les ordonnees des noeuds de conformation'
          read(*,*) (y1(i), i=1,nbrnde1)
*          write(8,*) ' ordonnees des noeuds de conformation (entree)'
*          write(8,*) (y1(i), i=1,nbrnde1)
        elseif(flag1.eq.1) then
          read(1,2000) ligne
          read(1,*)  (no(i), i=1,nbrnde)
          read(2,2000) ligne
          read(2,*)  (y(i),  i=1,nbrnde)
          read(7,2000) ligne
          read(7,*) (no1(i), i=1,nbrnde1)
          read(8,2000) ligne
          read(8,*) (y1(i),  i=1,nbrnde1)
          nbrnde2=0
          do 81 i=1,nbrnde
            if(y(i).lt.h2) nbrnde2=nbrnde2+1
 81       continue
          read(9,2000) ligne
          read(9,*) (no2(i), i=1,nbrnde2)
        else
          write(*,1006) flag1
          stop
        endif
      else
        write(*,1005) flag
        stop
      endif
*
      debit=((-2.0d0*gradp*(h1**3))/(3.0d0*etha0))*
     &      (1.0d0+(2.0d0/45.0d0)*(lambda*h1*gradp/etha0)**2)
      umoy=debit/(2.0d0*h2)
      umax=-0.5d0*(gradp/etha0)*(h1**2) - (lambda**2/54.0d0) *
     &     ((gradp/etha0)**3)*(h1**4)
      de=(umoy/h2)*lambda
      write(*,1249) umax
      write(*,1250) umoy
      write(*,1201) de
      write(3,1200) umoy
      write(3,1201) de
      do 20 i=1, nbrnde
        u$(i)=0.5d0*(gradp/etha0)*(y(i)**2-h1**2) +
     &        lambda**2/54.0d0*(gradp/etha0)**3 *
     &        (y(i)**4 - h1**4)
        u$(i)=u$(i)/umoy
```

```
        v$(i)=0.0
 20   continue
      do 21 i=1, nbrnde1
        sxy=gradp*y1(i)
        denom=27.0d0*etha0**2 + 2.0*lambda**2*sxy**2
        cxx$(i)=(9.0d0*etha0**2+ 2.0d0*lambda**2*sxy**2)/denom
        cyy$(i)=(9.0d0*etha0**2)/denom
        cxy$(i)=(3.0d0*etha0*lambda*sxy)/denom
 21   continue
*
* calcul du gradient de pression aval et du profil de vitesse
* a la sortie de la filiere
*
      if(dabs(h1-h2).le.1.0d-10) then
        gradp2=gradp
      else
        a=-2.0d0*h2**3/(3.0d0)
        b=lambda*h2
*
* estime initial du gradient de pression et iterations
*
        gradp2=(46.62d0*de**(-0.33333333d0))*(gradp/etha0)
 123    precedent=gradp2
        compteur=compteur + 1
*        write(*,*) ' gradp2/etha0 aval =',gradp2
        gradp2=-dsqrt((45.0d0/(2.0d0*b*b))*dabs(debit/(a*gradp2)-1.0d0))
        gradp2=(gradp2+precedent)/2.0d0
        erreur=dabs((gradp2-precedent)/gradp2)
        if(erreur.lt.1.0d-7) goto 124
        if(compteur.eq.100) goto  125
        goto 123
 125    write(*,*) ' pas de convergence gradp aval, erreur=',erreur
        stop
 124    write(*,*) ' gradp/etha0 aval = ', gradp2
        write(*,*) ' compteur=' ,compteur
        gradp2=gradp2*etha0
        tauxcis=-(lambda*h2*gradp2/etha0)*(1.0d0+(2.0d0/27.0d0)*
     &           (lambda*h2*gradp2/etha0)**2)
        write(*,*) 'taux de cisail. adim. paroi aval=',tauxcis
      endif
*
* calcul du profil de vitesses a la sortie
*
      do 70 i=1, nbrnde
        if(y(i).lt.h2) then
          u2$(i)=0.5d0*(gradp2/etha0)*(y(i)**2-h2**2) +
     &          lambda**2/54.0d0*(gradp2/etha0)**3 *
     &          (y(i)**4 - h2**4)
          u2$(i)=u2$(i)/umoy
        endif
 70   continue
*
      write(3,*) '** c.f. adimensionnelles sur U entree'
      do 30 i=1, nbrnde
        if(dabs(u$(i)).gt.1.0d-12) write(3,1500) no(i), u$(i)
 30   continue
*      if(compteur.ne. 100) then
      write(3,*) '** c.f. adimensionnelles sur U sortie'
      do 71 i=1, nbrnde2
        if(dabs(u2$(i)).gt.1.0d-12.and.y(i).lt.h2) then
          write(3,1500) no2(i), u2$(i)
        endif
 71   continue
*      endif
*      write(3,*) '** c.f. adimensionnelles sur V'
*      do 31 i=1, nbrnde
```

```
*        if(dabs(v$(i)).gt.1.0d-12) write(3,1500) no(i), v$(i)
* 31    continue
       write(3,*) '** c.f. adimensionnelles sur CXX'
       do 32 i=1, nbrnde1
        if(dabs(cxx$(i)).gt.1.0d-12) write(3,1500) no1(i), cxx$(i)
 32    continue
       write(3,*) '** c.f. adimensionnelles sur CYY'
       do 33 i=1, nbrnde1
        if(dabs(cyy$(i)).gt.1.0d-12) write(3,1500) no1(i), cyy$(i)
 33    continue
       write(3,*) '** c.f. adimensionnelles sur CXY'
       do 34 i=1, nbrnde1
        if(dabs(cxy$(i)).gt.1.0d-12) write(3,1500) no1(i), cxy$(i)
 34    continue
*
* calcul des profils de tenseurs de conformation a la sortie
* (pour verifier les resultats du code d'elements finis)
*
       if(compteur.ne.100) then
       do 51 i=1, nbrnde1
        no1(i)=no1(i)+25
        if(y1(i).le.h2) then
          sxy=gradp2*y1(i)
          denom=27.0d0*etha0**2 + 2.0*lambda**2*sxy**2
          cxx$(i)=(9.0d0*etha0**2+ 2.0d0*lambda**2*sxy**2)/denom
          cyy$(i)=(9.0d0*etha0**2)/denom
          cxy$(i)=(3.0d0*etha0*lambda*sxy)/denom
        endif
 51    continue
*
       write(3,*) '** CXX adimensionnelles a la sortie'
       do 52 i=1, nbrnde1
         if(y1(i).le.h2) then
           write(3,1503)  no1(i), cxx$(i)
         endif
 52    continue
       write(3,*) '** CYY adimensionnelles a la sortie'
       do 53 i=1, nbrnde1
         if(y1(i).le.h2) then
           write(3,1503)  no1(i), cyy$(i)
         endif
 53    continue
       write(3,*) '** CXY adimensionnelles a la sortie'
       do 54 i=1, nbrnde1
         if(y1(i).le.h2) then
           write(3,1503)  no1(i), cxy$(i)
         endif
 54    continue
       endif
*
 1003 format(' y(nbrdne)=',f12.8,'h=',f12.8,'devrait etre egaux')
 1005 format(' flag =',i3,' doit etre 0 ou 1')
 1006 format(' flag1 =',i3,' doit etre 0 ou 1')
 1200 format(' vitesse moyenne aval  =',f20.14)
 1201 format(' nombre de Deborah =',f20.14)
 1249 format(' umax amont='f20.14)
 1250 format(' umoy aval=',f20.14)
 1500 format(2x,i4,2x,f20.14)
 1501 format(4x,f20.14)
 1502 format(2x,f20.14)
 1503 format(2x,i4,2x,f20.14)
 2000 format(a75)
 5000 format(' voulez-vous lire les informations sur les noeuds'/
     &       ' de la section d"entree a partir de fichiers?'/
     &       ' oui=1, non=0')
      stop
```

```
end
```